Judgment Prediction via Injecting Legal Knowledge into Neural Networks

Leilei Gan, Kun Kuang*, Yi Yang and Fei Wu*
College of Computer Science and Technology, Zhejiang University, China
{leileigan, kunkuang, yangyics, wufei}@zju.edu.cn

Abstract

Legal Judgment Prediction (LJP) is a key problem in legal artificial intelligence, which aims to predict a law case's judgment based on a given text describing the facts of the law case. Most of previous works treat LJP as a text classification task and generally adopt deep neural networks (DNNs) based methods to solve it. However, existing DNNs based models are data thirsty and hard to explain which legal knowledge is based on to make such a prediction. Thus, injecting legal knowledge into neural networks to interpret the model and improve performance remains a significant problem. In this paper, we propose to represent declarative legal knowledge as a set of first-order logic rules and integrate these logic rules into a co-attention network-based model explicitly. The use of logic rules enhances neural networks with direct logical reasoning capabilities and makes the model more interpretable. We take private loan scenario as a case study and demonstrate the effectiveness of the proposed method through comprehensive experiments and analyses conducted on the collected dataset.

Introduction

Recently, applying artificial intelligence (AI) technologies to the legal domain has drawn more and more attention from the AI community, which can help legal practitioners to reduce heavy and repeat work from many aspects, such as legal judgment prediction, legal summarization, and legal question answering(Ye et al. 2018; Zhong et al. 2018; Duan et al. 2019; Zhong et al. 2020; Zhong et al. 2020).

Legal judgment prediction (LJP) is one of the most attractive research topic among these tasks (Xiao et al. 2018; Yang et al. 2019; Zhong et al. 2020). The goal of LJP is to predict a law case's judgment based on a given text, which describes the finding facts of the law case. Most of previous works treat LJP as a text classification task and generally adopt deep neural networks (DNNs) based methods to solve it. Zhong et al. (2018) and Yang et al. (2019) propose to use multi-task learning to capture the dependencies among subtasks by considering their topological order. Zhong et al. (2020) use a question answering task to improves the interpretability of LJP by minimizing the questions needed to ask through reinforcement learning.

*Corresponding Authors

Facts	On June 29, 2015, the defendant XXX borrowed 350,000 RMB yuan from the plaintiff XXX, and issued a loan note, which was agreed to be returned on August 29, 2015, at a monthly interest rate of 2%. After the expiry of the loan, the defendant did not repay the loan, ...
Claims	C1: The defendant returned the plaintiff's loan ... and paid interest on that amount at the rate of 2% per month ... Judge: Support

a) An example of legitimate interest claim

Facts	On June 10, 2013, the defendant XXX borrowed 10,000 RMB yuan from the plaintiff XXX, and issued a loan note at a monthly interest rate of 3%. After the expiry of the loan, the defendant did not repay the loan, ...
Claims	C1: The defendant returned the plaintiff's loan ... and paid interest on that amount at the rate of 3% per month ... Judge: Reject

b) An example of illegal claim for interest

Figure 1: Two examples in private loan law cases to show the importance of legal knowledge for predicting judgments. Each example consists of fact description, multiple claims proposed by the plaintiff, and a judgment made by judges.

Despite the success of applying deep neural networks to legal judgment prediction, most of the current methods are not combined well with legal knowledge, which distinguishes legal experts (e.g., lawyers and judges) from ordinary people. Examples in Figure 1 demonstrate the importance of legal knowledge for making correct judgment predictions. In Figure 1 (a), claim 1 (C1) proposed by the plaintiff about returning principal and paying interest at the rate of 2% per month is supported by the judge. However, in Figure 1 (b), a similar claim about paying interest at the rate of 3% per month is refused by the judge. The reason for making completely opposite judgments is that the claimed interest rate in Figure 1 (b) exceeds four times the quoted interest rate on the one-year loan market at the time the contract was established, which is not protected by the law[1]. However, the interest rate in Figure 1 (a) is legitimate. It is not trivial for non-legal experts or neural networks to make correct judgment predictions without knowing such legal knowledge.

Although a few works have attempted to integrate legal knowledge into neural networks (Luo et al. 2017; Xu et al.

[1]http://www.court.gov.cn/fabu-xiangqing-15146.html

2020). However, these works make use of legal knowledge through attention mechanisms or graph neural networks are coarse-grained and implicit, which can not enhance neural networks with logical reasoning capabilities directly.

To teach neural networks legal knowledge explicitly, we propose to combine the DNNs with a symbolic legal knowledge module, which contains a set of first-order logic (FOL) rules. Using FOL to represent domain knowledge has already demonstrated its effectiveness on many other tasks, including visual relation prediction (Xie et al. 2019), natural language inference (Li et al. 2019), and semantic role labeling (Li et al. 2020). The advantages of representing legal knowledge as first-order logic rules have two folds. First, it makes judgment prediction more interpretable, which is critical in the legal domain. Second, logic rules provide models with inductive bias, which reduces the dependency of neural networks on data. To the best of our knowledge, we are the first to combine neural networks with legal knowledge expressed as FOL rules.

Our proposed model unifies the gradient-based deep learning module with the non-differentiable symbolic knowledge module via probabilistic logic, as shown in Figure 2. Specifically, the deep learning module is first built based on a co-attention mechanism, which can benefit the information interaction between fact descriptions and claims. Afterwards, the outputs of deep learning module, predicted probability distribution for judgments, will be fed into the symbolic module. The logic rules in the symbolic module then adjust the probability distribution to avoid outputs violating the law. For example, a FOL rule, $\neg X_{\text{RIG}} \wedge X_{\text{DIA}} \wedge X_{\text{TIR}} \rightarrow \neg Y$, meaning that "interest rate exceeding four times the quoted interest rate on the one-year loan market is not protected by the law", will decrease the score of supporting the claims for interests. Another obstacle to unifying the two modules is the non-differential characteristic of FOL rules. To make the unified model can be trained in an end-to-end way, we define some mapping functions to convert the discrete outputs of logic rules into continuous real-values.

We take the private loan scenario as a case study, which is not trivial for DNNs to predict the judgments, due to many numbers and dates often appearing in the claims and facts texts. The effectiveness of the proposed method is evaluated through comprehensive experiments and analyses conducted on the collected datasets.

To summarize, our contributions are:

1. We investigate the importance of legal knowledge in the private loan scenario and collect the first large private loan judgment prediction dataset.
2. We formulate the legal knowledge as a set of first-order logic rules and integrate these symbolic rules into a co-attention network-based model.
3. We evaluate the proposed method in the collected private loan dataset through extensive experiments and analyze the role legal knowledge plays.

Note that one advantage of our method is that it can inject legal knowledge into DNNs without adding additional training parameters.

Related Work

Legal Judgment Prediction. Legal judgment prediction (LJP) is one of the most attractive research topic in the legal artificial intelligence area (Zhong et al. 2020; Xiao et al. 2018). Generally, most existing works apply various deep learning methods, e.g., CNNs or LSTMs, to encoder the fact descriptions and make predictions through a classification layer. Zhong et al. (2018) and Yang et al. (2019) propose to use multi-task learning to capture the dependencies among by considering the topological order between the subtasks. Zhong et al. (2020) uses a question answering task to improves the interpretability of LJP by using reinforcement learning to minimize the questions needed to ask.

Since legal knowledge is critical for making judgments, a line of works have already attempted to integrate legal knowledge into neural networks to improve performance. Luo et al. (2017) formulate legal articles as a knowledge basis and use attention mechanisms to aggregate the representations of relevant legal articles to support the judgment prediction. Hu et al. (2018) manually annotate discriminative legal attributes for confusing charges and incorporate this kind of explicit knowledge into an attention-based multi-task learning judgment prediction framework. Xu et al. (2020) explore to extract distinguish knowledge from similar law articles using a graph-based method.

However, prior works making use of legal knowledge through attentions or graph neural networks are implicit and unexplainable. In this paper, we investigate to represent legal knowledge as first-order logic rules, which are more explicit and provide more interpretability.

Logic in Neural Networks. Combining deep neural networks with symbolic knowledge, which often is represented as first-order logic rules, has emerged as one promising topic in the AI community (Xiao, Dymetman, and Gardent 2017; Manhaeve et al. 2018; Dong et al. 2019; Reimann and Schwung 2019; Li and Srikumar 2019). Hu et al. (2016) construct teacher networks with logic rules as regularization and then distill rule knowledge into student networks iteratively. Xu et al. (2018) design a semantic loss function, which can satisfy a given logical constraint. Xie et al. (2019) propose LENSR which expresses symbolic knowledge in deterministic decomposable negation normal form, and learns semantically-constrained embeddings with graph neural networks. Li et al. (2020) and Wang and Pan (2020) propose to constraint the outputs of neural networks with structure knowledge representing as first-order logic rules in semantic role labeling and relation extraction tasks, respectively. Li et al. (2019) and Minervini and Riedel (2018) investigate the inconsistency problem in the natural language inference task and propose to use first-order logic based background knowledge to improve consistency.

To the best of our knowledge, we are the first to combine symbolic knowledge with neural networks using first-order logic rules in the legal artificial intelligence domain.

Methodology

We first introduce some notations and formulate the legal judgment prediction task in the private loan scenario.

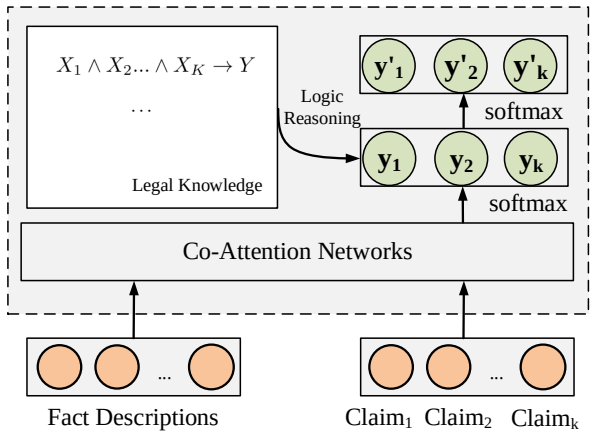

Figure 2: The overall architecture.

Let $F = \{w_1, w_2, ..., w_N\}$ denotes the fact description of a law case, where $w_i \in V$ represents a word and N is the word sequence length. V is the fixed vocabulary. Let $C = \{c_1, c_2, ..., c_K\}$ denotes K claims proposed by the plaintiff, where $c_i = \{w_{i1}, w_{i2}, ..., w_{iM}\}$ represents a word sequence of length M, where $w_{ij} \in V$. Given the fact description F and K claims C, the task aims to predict corresponding $y_i \in Y$ for each c_i in C.

As shown in Figure 2, the proposed model consists of a deep learning module based on co-attention networks and a symbolic legal knowledge module. We first input the word representations of fact descriptions and multiple claims into the co-attention network to obtain contextual representations for both fact descriptions and claims. Then, the predicted probability distribution of the deep learning module is re-weighted by first-order logic rules in the symbolic module. The logic rules represent professional legal knowledge which is essential for making correct judgments.

Co-Attention Networks

When a judge considers whether to support the claims or not, she or he should first retrieve related parts from the fact description according to the claims. Correspondingly, which parts of the claims are significant for the prediction, such as the interest rate in the claims, should also be paid more attention. Inspired by this procedure, a bi-directional attention network is used to enrich the representations by exchanging information between fact descriptions and claims.

Word Embedding Layer. In this layer, given a fact description F and claims C which has K different but correlated claims, a pre-trained word embedding layer is first used to obtain word vectors for each word as follows:

$$\mathbf{x}_f = \mathbf{Emb}(F) \in \mathbb{R}^{N \times d} \quad (1)$$

$$\mathbf{x}_c = \mathbf{Emb}(C) \in \mathbb{R}^{K \times M \times d}, \quad (2)$$

where $\mathbf{Emb}$ is the pre-trained word embeding layer and d is the size of word vectors.

To simplify the notation, we omit K for remaining parts of this paper.

Contextual Representation Layer. In this layer, a bi-directional long short term memory network (BiLSTM;(Hochreiter and Schmidhuber 1997)) is used to capture contextual representations of sequences for both fact descriptions and claims as following:

$$\mathbf{H}_f = \mathbf{BiLSTM}(\mathbf{x}_f) \in \mathbb{R}^{N \times h} \quad (3)$$

$$\mathbf{H}_c = \mathbf{BiLSTM}(\mathbf{x}_c) \in \mathbb{R}^{M \times h}, \quad (4)$$

where h is the hidden size of BiLSTM.

Attention Layer. We use a co-attention mechanism to align relevant factual parts between claims and fact descriptions in this layer, which has two attention directions: from claims to fact descriptions and from fact descriptions to claims. The two attention directions thereafter lead to claims-aware representations of facts and facts-aware representations of claims, respectively.

Specifically, we first conduct soft alignment between fact descriptions and claims by using dot product to calculate word to word similarities between $\mathbf{H}_c$ and $\mathbf{H}_f$ as following:

$$\mathbf{S} = \mathbf{H}_f \cdot \mathbf{H}_c^T \in \mathbb{R}^{N \times M}.$$

Then, we use the attention direction from claims to facts description to obtain claims-aware fact representations. We apply a softmax function on S to weight which words in the claims are significant to each word in the fact descriptions as following:

$$\alpha_i = \mathbf{softmax}(\mathbf{S}, dim = 1). \quad (5)$$

Then each row in the claims-aware fact representations $\widetilde{\mathbf{H}}_f \in \mathbb{R}^{N \times h}$ is the weighted sum of rows in claims representations:

$$\widetilde{\mathbf{H}}_f^i = \sum_i \alpha_i \mathbf{H}_c^i \in \mathbb{R}^h. \quad (6)$$

Similarly, we use the other attention direction to obtain facts-aware claims representations, which can pay more attention to relevant parts in the facts which are significant to the claims as following:

$$\beta_i = \mathbf{softmax}(\mathbf{S}, dim = 2)$$

$$\widetilde{\mathbf{H}}_c^i = \sum_i \beta_i \mathbf{H}_f^i \in \mathbb{R}^h,$$

where $\widetilde{\mathbf{H}}_c^i$ is the row vector of the facts-aware claim representations.

We fuse $\mathbf{H}_c$, $\mathbf{H}_f$, $\widetilde{\mathbf{H}}_c$ and $\widetilde{\mathbf{H}}_f$ by concatenating these contextual representations as following:

$$\mathbf{G} = [\mathbf{H}_c, \widetilde{\mathbf{H}}_f, |\mathbf{H}_c - \widetilde{\mathbf{H}}_f|, \mathbf{H}_c \circ \widetilde{\mathbf{H}}_c], \quad (7)$$

where $\circ$ is the element-wise product operation.

Output Layer. Finally, the fused representations G is fed into a fully connected network with a softmax activation function to output the predicted probability distributions:

$$\mathbf{y} = \mathbf{softmax}(\mathbf{W_p G}), \quad (8)$$

where W_p is trainable model parameters.

Note that the softmax outputs of co-attention networks will be input into the logic module and be adjusted accordingly.

Legal Knowledge as First-Order Logic Rules

As discussed in the prior section, the introduced co-attention model can fuse the representations of claims and fact descriptions to make implicit reasoning. However, the related legal knowledge used by legal experts (e.g., lawyers or judges) can hardly be learned by the co-attention network. For example, the rule that the interest rate of private loans exceeding 2% per month is not protected by law may not always be followed by the neural networks. Thus, it is crucial to inject such declarative legal knowledge explicitly into neural networks to make correct and interpretable judgment predictions.

Before presenting how to integrate legal knowledge into DNNs, we first introduce first-order logic which is used to represent legal knowledge briefly.

First-Order Logic. FOL is an expressive logical system to represent domain knowledge. Formally, the FOL system consists of constants, variables, predicts, and several propositional connectives, including conjunction ($\wedge$), disjunction ($\vee$), negation ($\neg$) and quantifiers (e.g., $\exists$ and $\forall$). Constants and variables are denoted as lower-case and upper-case letters, respectively. In this paper, we take the simple conditional statement in FOL to represent legal knowledge, which is formulated as $X \rightarrow Y$, where X and Y is called precondition and consequent, respectively. Precondition can be conjunctions or disjunctions of variables, e.g., $X_1 \wedge ... \wedge X_K \rightarrow Y$. The grounding of a formula $X \rightarrow Y$ is to substitute each variable in the precondition and consequent with constants.

However, the original consequent Y of FOL rule is not differentiable, which cannot be combined with the deep learning module directly. To preserve the advantages of gradient-based end-to-end training schema, we convert the Boolean operations of FOL into probabilistic logic, which is denoted on the continuous real-valued space.

Specifically, we associate the variable X in preconditions with corresponding neural outputs x. Then, Łukasiewicz T-norm and T-conorm (Klement, Mesiar, and Pap 2000) are used to relax the logic rules into soften version based on the associated outputs of the deep learning module. We follow Li and Srikumar (2019) to denote a set of functions, which are used to map the discrete outputs of FOL into continuous real values as following:

- $\Gamma(X_i) = x_i$ with X_i denoting a variable in FOL and x_i as the associate output of neural networks.
- $\Gamma(\bigwedge_i X_i) = \max(0, \sum_i x_i - |X| + 1)$.
- $\Gamma(\bigvee_i X_i) = \min(1, \sum_i x_i)$.
- $\Gamma(\neg \bigvee_i X_i) = \max(0, 1 - \sum_i x_i)$.
- $\Gamma(\neg \bigwedge_i X_i) = \min(0, N - \sum_i x_i)$.

The first principle to design qualified mapping functions is that when the precondition holds, the mapping function should generate a predefined maximum positive score to lift the original score produced by neural networks. Second, the mapping functions should reveal the semantics of propositional connectives as well. For example, the mapping score of a conjunctive precondition becomes zero if even only one of the conjuncts is false. For a disjunctive precondition, the mapping score becomes zero when all the disjuncts are false. Moreover, the mapping score will increase as the number of disjuncts being true increases.

In addition to the functions listed above, two mapping functions are also used for negated predicates. One of them is for negated predicates in preconditions, e.g., $\neg X_i$. The soften output of $\neg X_i$ is denoted as $1 - x_i$. The other is for negated consequent $\neg Y$, which is denoted as $-y_i$ with the purpose of reducing the original outputs of neural networks.

Injecting Legal Knowledge into DNNs. Before introducing concrete legal knowledge related to our private loan scenario, we first show how to inject symbolic FOL rules into the deep learning module by using the above mapping functions $\Gamma(\cdot)$. In short, the core idea of this legal knowledge injection is to re-weight the output $\mathbf{y}$ of co-attention networks as introduced in the previous subsection so that when the facts in the text satisfy conditions in the legal knowledge, the associated value of $\mathbf{y}$ increases. Otherwise, the value of $\mathbf{y}$ decreases.

Specifically, given the softmax outputs $\mathbf{y}$ of Eq. (8) and a FOL rule $X \rightarrow Y$, the FOL rule and DNNs are combined by regulating the outputs of deep learning module as following:

$$\mathbf{y}' = \mathbf{softmax}(\mathbf{y} + \rho\Gamma(X)), \tag{9}$$

where ρ is a hyper-parameter which denotes the importance of each rule.

The motivation of designing the above function is that by first compiling declarative legal knowledge into FOL rules and then using mapping functions to convert symbolic rules into continuous real values, we can directly regulate the outputs of deep learning module. Note that the final predictions $\mathbf{y}'$ is decided by both the deep learning module and symbolic knowledge module.

Legal Knowledge. We investigate to compile three typical legal knowledge into FOL rules, which is frequently referred by legal experts in private loan cases.

The first legal logic rule comes from Article 28 of the Provisions of the Supreme People's Court on Several Issues Concerning the Application of Law in the Trial of Private Loan Cases[2]. In short, this article states that the interest rate agreed by the lender and the borrower exceeding four times the quoted interest rate on the one-year loan market at the time the contract was established shall not be supported by the law. We formulate this legal knowledge as the following FOL rule K1:

$$X_{\text{TIR}} \wedge X_{\text{RIO}} \rightarrow \neg Y \tag{10}$$

where X_{TIR} is a variable representing if the current claim is for interest. X_{RIO} means if the claimed interest rate exceeds the four times the quoted interest rate on the one-year loan market. This rule will decrease the score of support the claim for illegitimate interest rate.

The second legal logic rule comes from Article 29 of the same law which rule K1 is base on. In short, it states that if

[2]http://www.court.gov.cn/fabu-xiangqing-15146.html

Split	Support	Partially Support	Reject
Training Set	70,386	18,921	6,438
Validation Set	8,777	2,440	858
Test Set	8,839	2,293	855

Table 1: Statistics of private loan dataset

neither the interest rate during the loan period nor the overdue interest rate has been agreed upon, the overdue interest from the date of overdue repayment shall be supported by the court. We formulate this legal knowledge as the following FOL rules K2:

$$X_{\text{TIR}} \wedge \neg X_{\text{RIA}} \wedge \neg X_{\text{DIL}} \rightarrow \neg Y \tag{11}$$

where X_{RIA} is a variable representing if the borrower and the lender have made an agreement on the interest rate or not. X_{DIL} means if the date of overdue repayment is legitimate.

In the private loan law cases, the plaintiff often proposes multiple claims, and the judgments of these claims are not independent. For example, if a plaintiff proposes two claims and one of the claims is for principal and the other is for interest. If the judge does not support the principal claim, then the interest claim should not be supported either. Such prior knowledge should be injected into the deep learning module as well. Another example showing the dependency within multiple claims is that the litigation costs shall be borne by the losing party. This rule comes from Article 29, Chapter V of the Measures on Litigation Fees of the People's Courts [3]. The third FOL rule K3 is formulated as:

$$\bigwedge_{j \in s, j \neq i} Y_j \wedge X_{\text{TIC}} \rightarrow Y_i, \tag{12}$$

where X_{TIC} means if the current claim is for litigation fees or not. This rule will affect those claims for litigation costs. For example, if a plaintiff has three claims and two of the claims are supported by the judge, then the last claim for litigation cost will be predicted as support according to this legal knowledge.

Note that the values of the variables in precondition are not labelled in the original dataset. We use heuristic rules to extract their corresponding values in each instance.

Training

Given a set of samples, $D = \{F_i, C_i^j|_{j=1}^K\}|_{i=1}^T$, the model is trained by maximizing the following objective function:

$$J = \sum_{i=1}^{T} \sum_{j=1}^{K} \mathbf{log}(y'_{ij}). \tag{13}$$

Experiments

In this section, we compare our method with other deep learning-based baselines on a collected private loan dataset, discussing the role the legal knowledge playing in the performance.

[3]http://www.court.gov.cn/jianshe-xiangqing-5092.html

Parameter	Value	Parameter	Value
Word emb size	300	Dropout	0.2
BERT emb size	768	Batch size	16
LSTM layer	1	Learning rate decay	0.05
LSTM hidden	256	Early Stoping	10

Table 2: Hyper-parameters

Settings

In the experiments, we collected a total of 61, 611 private loan law cases. Each instance in the dataset consists of a fact description and the plaintiff's multiple claims. On average, each case contains 5.94 sentences in the fact description and 1.89 claims. The claim is labeled as Support, Partially Support or Reject. Statistics of the dataset is shown in Table 1. To the best of our knowledge, this is the very first large private loan judgment prediction dataset. We will release all the experiment data to motivate other scholars to further investigate this problem[4].

We use the Skip-Gram model (Mikolov et al. 2013) to train word embeddings on the judgment documents. The dimension of word embeddings is set to 300. The size of hidden states of bidirectional-LSTM is 256. The neural networks are trained using Adam Optimization (Kingma and Ba 2014) with a learning rate set to 0.001, and perform the mini-batch gradient descent with a batch size of 16. For BERT, the learning rate and batch size are set to 5e-6 and 1, respectively. We use Macro F1 and Micro F1 (Mac.F1 and Mic.F1 for short) as the main metrics for algorithm evaluation. An early stopping strategy is used that if the sum of Mac.F1 and Mic.F1 on the development dataset does not increase for ten epochs, the training process is terminated. Table 2 shows the values of model hyper-parameters.

Baselines

We compare our model with other strong baselines, ranging from traditional machine learning methods to advanced pre-trained models.

TF-IDF+SVM is a robust multi-class classification model based on support vector machine (SVM;(Suykens and Vandewalle 2004)) using TF-IDF features.

TextCNN is a convolutional neural network trained on top of pre-trained word vectors for sentence-level classification tasks (Kim 2014) where the entire fact description and claims are concatenated as input.

BiLSTM+ATT employs Bi-directional LSTM with attention mechanism to capture context semantics and automatically selects important features through attention during training, which is a variant of attention-based RNNs.

HARNN stands for Hierarchical Attention Network (Yang et al. 2016) which is a hierarchical document classification model with two levels of attention mechanisms for aggregating words to sentences and sentences to documents.

[4]Code and dataset will be publicly available at https://github.com/leileigan/LawReasoning.

Method	Reject			Partially Support			Support			Average			
	P	R	F1	P	R	F1	P	R	F1	Mac.P	Mac.R	Mac.F1	Mic.F1
TF-IDF+SVM	75.1	49.4	59.6	58.1	45.4	51.0	84.8	92.2	88.4	**80.5**	66.3	62.4	72.7
TextCNN	75.6	43.4	55.1	66.5	41.7	51.3	83.0	94.5	88.4	75.0	59.9	64.9	80.7
BiLSTM+ATT	72.2	52.8	60.9	64.5	51.2	57.1	85.8	**92.6**	89.0	74.1	65.5	69.0	81.8
HARNN	**75.8**	52.1	61.7	63.3	50.5	56.2	85.5	92.3	88.9	74.9	65.0	68.9	81.6
BERT	72.3	60.8	66.7	64.5	57.9	61.0	87.6	91.3	89.4	74.8	70.0	72.2	82.7
RoBERTa	71.7	63.4	67.3	66.8	58.1	62.1	87.9	91.9	89.9	75.1	71.1	73.1	83.4
AutoJudge	75.7	67.2	71.3	70.5	71.9	71.2	91.2	91.7	91.5	79.2	77.0	77.9	86.2
CoATT	70.5	**72.8**	71.6	**72.7**	69.0	70.8	91.4	92.3	91.8	78.2	78.0	78.1	86.4
CoATT + LK	75.0	69.5	**72.1**	71.8	**75.3**	**73.5**	**92.6**	92.1	**92.3**	79.8	**78.9**	**79.3**	**87.2**

Table 3: Final results of all methods on civil loan test dataset. The bold results are the best results.

%Train	Method	Mic-F1	Mac-F1	Gains
1%	CoATT	77.93	60.46	-
1%	CoATT + LK	77.25	63.02	↑ 1.88
5%	CoATT	79.81	65.81	-
5%	CoATT + LK	79.79	67.62	↑ 1.42
10%	CoATT	81.56	68.94	-
10%	CoATT + LK	81.39	69.97	↑ 0.86

Table 4: The effect of different corpus size.

Method	Mic-F1	Mac-F1	Gains
CoATT	86.44	78.09	-
CoATT + K1	86.73	78.43	↑0.63
CoATT + K2	86.84	78.40	↑0.73
CoATT + K3	86.76	78.33	↑ 0.56
CoATT + ALL	87.24	79.31	↑2.02

Table 5: The effect of different rules.

AutoJudge (Long et al. 2019) propose to model the complex interactions between claims and fact descriptions via pairwise attention.

BERT (Devlin et al. 2019) and **RoBERTa** (Liu et al. 2019) are pre-trained contextual representations model. We concatenate fact description, "[SEP]" and claim as input and take the representation of "[CLS]" as aggregated representations for the final judgment prediction. We use the version of 12 heads and 12 layers for both BERT and RoBERTa [5].

We denote the co-attention based method as "CoATT". "+LK" means we inject legal knowledge into neural networks.

Overall Performance. We evaluate our model and the baselines on the private loan dataset. In addition to Mac.F1 and Mic.F1, we also use macro-precision (Mac.P) and macro-recall (Mac.R) to evaluate these methods. The performance on the test set is summarized in Table 3. We can draw the following conclusions from the results: First, the deep learning-based methods, e.g., TextCNN, BiLSTM+ATT, and HARNN, exceed the traditional machine learning method TF-IDF+SVM by a large, which shows the success of applying neural networks for LJP. Second, LSTM based methods give better results than CNN based method, which demonstrates the advantages of extracting contextual features using LSTM. Third, BERT outperforms all the deep learning-based methods, which shows the strong representation abilities of the pre-trained language model, even for the legal domain. Finally, the proposed co-attention model gives a 4.8% absolute increase in performance (the average of Mac.F1 and Mic.F1) compared with BERT, which leads to two further conclusions. First, directly applying pre-trained models to specific domains still has room for improvement. Second, it verifies our assumption that the bi-directional attention flows of information between facts and claims help to locate crucial facts. Most importantly, injecting legal knowledge into co-attention networks gives another 1% absolute increase compared with the co-attention model and achieves the best results among all methods.

Low Resource Scenarios. We conduct a set of experiments with different training data size to investigate the performance of our method in low resource scenarios. The aim of this set of experiments is to answer whether injecting legal knowledge into DNNs can provide inductive bias and reduce the data dependency for training. As shown in Table 4, combining neural networks with legal knowledge shows improvement in all settings, which demonstrates the effectiveness of our method. Moreover, with the decrease in data size, the improvement increases in general. The 1% training data size setting gives the biggest gains. We regard that the smaller the training data size is, the harder it is for neural networks to learn implicit logical reasoning ability from data. However, injecting prior knowledge provides neural networks with inductive bias, which reduces data thirsty.

Effect of Different Rules. To demonstrate the effectiveness of each legal rule, we conduct ablation studies on different model settings. The results are listed in Table 5. First of all, we can conclude that co-attention networks combined with different logic rules give better performance than co-attention networks without using legal knowledge. Second, injecting all three logic rules into neural networks gives the best result. And the gain is bigger than the sum of the gains

[5] https://github.com/huggingface/pytorch-pretrained-BERT

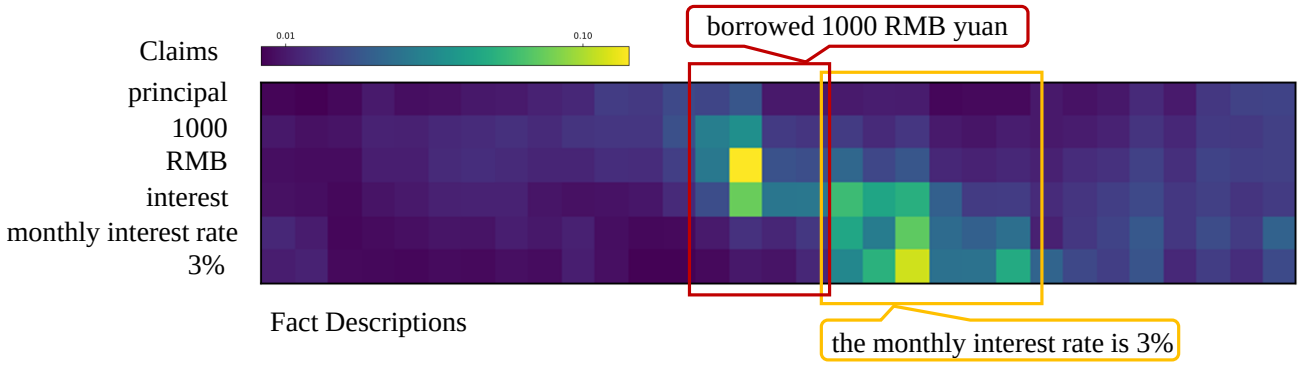

Figure 3: The heat map shows the Co-Attention matrices between claims and fact descriptions. Each row is a select key word in claims, e.g., principal and interest rate. Each column in the heat map is a context word from fact descriptions.

when using these logic rules solely. We assume that these logic rules influence each other mutually, which leads to better results.

Convergence Comparison. We conduct experiments on the development set of private civil loan dataset to compare the convergence speeds of different models. As shown in Figure 4, compared with TextCNN, BiLSTM based methods show a higher starting point and increase steadily against the iterations. Second, while BERT gives better results compared with deep learning methods, its performance decreases instead as the number of epoch increases. Third, co-attention methods with or without legal knowledge demonstrate competitive performance compared with BERT in the first few iterations. However, our methods can still gain improvement when the number of epoch increases, showing advantages over BERT. Finally, when injecting legal knowledge into co-attention models, the CoATT+LK achieves the best performance and has a smaller decrease compared with CoATT when the number of epoch increases.

Case Study. We dive into some representative examples to give an intuitive illustration of how co-attention networks.

As shown in Figure 3, the heat map shows the co-attention matrices for claims and fact descriptions. Each row in the heat map is a concerned word in claims, e.g., principal and interest rate. Each column is a context word from fact descriptions. In sum, the selected words from claims match two different regions in the fact description. For example, words related to interest, e.g., monthly interest rate and 3%, in the claim have higher attention scores for those words describing finding facts about interest. This phenomenon demonstrates that co-attention networks can associate different claims parts to related finding facts, which is beneficial to verify the true or false of the claims proposed by the plaintiff.

Conclusion

In this work, we investigate how to inject legal knowledge into legal judgment prediction explicitly. The proposed model represents declarative legal knowledge as a set of first-order logic rules and integrate these logic rules into a co-attention network based model in an end-to-end way. The use of logic rules enhances the neural networks with direct logical reason capabilities and makes the model more interpretable. Moreover, the inductive bias introduced by legal knowledge relieves the thirst of deep neural networks for data. Our method is evaluated on a private loan dataset and show its advantages over other baselines through extensive experiments.

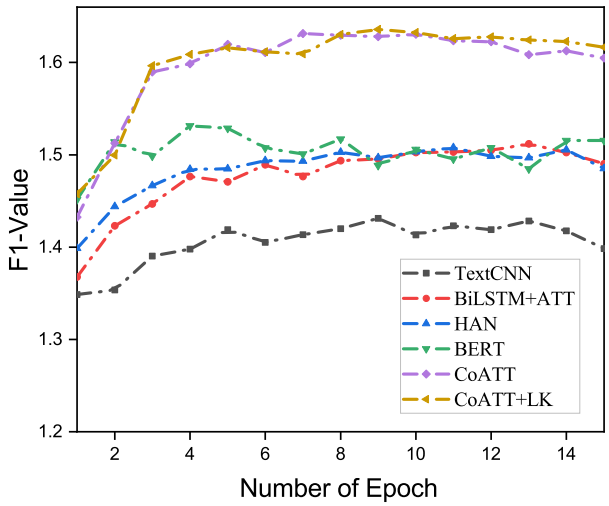

Figure 4: F1-value of all methods on civil load development dataset against the number of epoch.

Acknowledgements

This work was supported in part by Key Research and Development Projects of the Ministry of Science and Technology of China (No. 2020YFC0832500), National Natural Science Foundation of China (No. 62006207), National Key Research and Development Program of China (No. 2018AAA0101900), the Fundamental Research Funds for the Central Universities and Zhejiang Province Natural Science Foundation (No. LQ21F020020).

References

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.

Dong, H.; Mao, J.; Lin, T.; Wang, C.; Li, L.; and Zhou, D. 2019. Neural Logic Machines. In *ICLR 2019 : 7th International Conference on Learning Representations*.

Duan, X.; Zhang, Y.; Yuan, L.; Zhou, X.; Liu, X.; Wang, T.; Wang, R.; Zhang, Q.; Sun, C.; and Wu, F. 2019. Legal Summarization for Multi-role Debate Dialogue via Controversy Focus Mining and Multi-task Learning. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 1361–1370.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation* 9: 1735–1780.

Hu, Z.; Li, X.; Tu, C.; Liu, Z.; and Sun, M. 2018. Few-Shot Charge Prediction with Discriminative Legal Attributes. In *COLING 2018: 27th International Conference on Computational Linguistics*, 487–498.

Hu, Z.; Ma, X.; Liu, Z.; Hovy, E. H.; and Xing, E. P. 2016. Harnessing Deep Neural Networks with Logic Rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, 2410–2420.

Kim, Y. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746–1751.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Klement, E.; Mesiar, R.; and Pap, E. 2000. Triangular Norms. In *Trends in Logic*.

Li, T.; Gupta, V.; Mehta, M.; and Srikumar, V. 2019. A Logic-Driven Framework for Consistency of Neural Models. In *2019 Conference on Empirical Methods in Natural Language Processing*, 3922–3933.

Li, T.; Jawale, P. A.; Palmer, M.; and Srikumar, V. 2020. Structured Tuning for Semantic Role Labeling. In *ACL 2020: 58th annual meeting of the Association for Computational Linguistics*, 8402–8412.

Li, T.; and Srikumar, V. 2019. Augmenting Neural Networks with First-order Logic. In *ACL 2019 : The 57th Annual Meeting of the Association for Computational Linguistics*, 292–302.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* .

Long, S.; Tu, C.; Liu, Z.; and Sun, M. 2019. Automatic judgment prediction via legal reading comprehension. In *China National Conference on Chinese Computational Linguistics*, 558–572. Springer.

Luo, B.; Feng, Y.; Xu, J.; Zhang, X.; and Zhao, D. 2017. Learning to Predict Charges for Criminal Cases with Legal Basis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2727–2736.

Manhaeve, R.; Dumancic, S.; Kimmig, A.; Demeester, T.; and De Raedt, L. 2018. Deepproblog: Neural probabilistic logic programming. *Advances in Neural Information Processing Systems* 31: 3749–3759.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed Representations of Words and Phrases and their Compositionality. *ArXiv* abs/1310.4546.

Minervini, P.; and Riedel, S. 2018. Adversarially Regularising Neural NLI Models to Integrate Logical Background Knowledge. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, 65–74.

Reimann, J. N.; and Schwung, A. 2019. Neural Logic Rule Layers. *arXiv preprint arXiv:1907.00878* .

Suykens, J.; and Vandewalle, J. 2004. Least Squares Support Vector Machine Classifiers. *Neural Processing Letters* 9: 293–300.

Wang, W.; and Pan, S. 2020. Integrating Deep Learning with Logic Fusion for Information Extraction. *AAAI 2020 : The Thirty-Fourth AAAI Conference on Artificial Intelligence* 34(5): 9225–9232.

Xiao, C.; Dymetman, M.; and Gardent, C. 2017. Symbolic Priors for RNN-based Semantic Parsing. In *Twenty-Sixth International Joint Conference on Artificial Intelligence*, 4186–4192.

Xiao, C.; Zhong, H.; Guo, Z.; Tu, C.; Liu, Z.; Sun, M.; Feng, Y.; Han, X.; Hu, Z.; Wang, H.; and Xu, J. 2018. CAIL2018: A Large-Scale Legal Dataset for Judgment Prediction. *ArXiv* abs/1807.02478.

Xie, Y.; Xu, Z.; Kankanhalli, M. S.; Meel, K. S.; and Soh, H. 2019. Embedding Symbolic Knowledge into Deep Networks. In *Advances in Neural Information Processing Systems*, 4233–4243.

Xu, J.; Zhang, Z.; Friedman, T.; Liang, Y.; and den Broeck, G. V. 2018. A Semantic Loss Function for Deep Learning with Symbolic Knowledge. In *ICML 2018: Thirty-fifth International Conference on Machine Learning*, 5498–5507.

Xu, N.; Wang, P.; Chen, L.; Pan, L.; Wang, X.; and Zhao, J. 2020. Distinguish Confusing Law Articles for Legal Judgment Prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 3086–3095. Online: Association for Computational Linguistics. doi:10.18653/v1/2020.acl-main.280. URL https://www.aclweb.org/anthology/2020.acl-main.280.

Yang, W.; Jia, W.; Zhou, X.; and Luo, Y. 2019. Legal Judgment Prediction via Multi-Perspective Bi-Feedback Network. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 4085–4091.

Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; and Hovy, E. 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*, 1480–1489.

Ye, H.; Jiang, X.; Luo, Z.; and Chao, W. 2018. Interpretable Charge Predictions for Criminal Cases: Learning to Generate Court Views from Fact Descriptions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, 1854–1864.

Zhong, H.; Wang, Y.; Tu, C.; Zhang, T.; Liu, Z.; and Sun, M. 2020. Iteratively Questioning and Answering for Interpretable Legal Judgment Prediction. *AAAI 2020 : The Thirty-Fourth AAAI Conference on Artificial Intelligence* 34(1): 1250–1257.

Zhong, H.; Xiao, C.; Tu, C.; Zhang, T.; Liu, Z.; and Sun, M. 2020. How Does NLP Benefit Legal System: A Summary of Legal Artificial Intelligence. In *ACL*.

Zhong, H.; Xiao, C.; Tu, C.; Zhang, T.; Liu, Z.; and Sun, M. 2020. JEC-QA: A Legal-Domain Question Answering Dataset. *AAAI 2020 : The Thirty-Fourth AAAI Conference on Artificial Intelligence* 34(5): 9701–9708.

Zhong, H.; Zhipeng, G.; Tu, C.; Xiao, C.; Liu, Z.; and Sun, M. 2018. Legal Judgment Prediction via Topological Learning. In *EMNLP 2018: 2018 Conference on Empirical Methods in Natural Language Processing*, 3540–3549.