Expert Systems With Applications 238 (2024) 122177

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Chinese legal judgment prediction via knowledgeable prompt learning

Jingyun Sun [*], Shaobin Huang, Chi Wei

College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China

ARTICLE INFO

Keywords:
Legal AI
Legal judgment prediction
Prompt learning
Few-shot

ABSTRACT

In recent years, applying AI techniques in the legal field has attracted researchers' attention. In particular, Legal Judgment Prediction (LJP), which aims to predict accusations based on given case description texts, has attracted much attention from the natural language processing community. However, most of the existing LJP methods are data-intensive. As we know, data annotation in the legal field is expensive. Prompt learning is a recently prevalent methodology, which often achieves surprising results in few-shot or even zero-shot scenarios. We propose a novel method for Chinese LJP based on prompt learning called KnowPrompt4LJP. The method aligns the Chinese LJP task with the pre-training task of a Pre-trained Language Model (PLM) via a prompt template to stimulate the PLM's recall of learned knowledge. In addition, the well-designed prompt template can enhance the PLM's understanding of the Chinese LJP task. We also use an external knowledge base to extract keyword information from the Chinese case description texts and incorporate it into the prompt template, thus enhancing the guidance of the prompt template to the PLM. Experimental results on CAIL2018, a high-quality Chinese LJP competition dataset, show that KnowPrompt4LJP achieves far better results than the baselines in zero-shot, few-shot, and full-size training data scenarios. KnowPrompt4LJP can achieve a macro F1 value of 0.70 in the low-resource scenario, which is comparable to the baselines' results in the data-rich scenario. In the scenario of using full-size training data, *KnowPrompt4LJP* can achieve a macro F1 value of 0.81.

1. Introduction

In recent years, applying artificial intelligence techniques to the legal field has attracted many researchers' attention. In particular, legal texts, the most dominant carrier of legal information, have stimulated research interest in the natural language processing community (Shaghaghian et al., 2020; Hendrycks et al., 2021). Related studies mainly include legal judgment prediction (Ma et al., 2021; Yue et al., 2021), similar case matching (Bhattacharya et al., 2022; Fang et al., 2022), legal information extraction (Hong et al., 2021, Mandal et al., 2021), legal Q&A system (Fawei et al., 2018, Zhong, Xiao et al., 2020), etc. On the one hand, these studies help reduce legal practitioners' workload and improve their work efficiency. On the other hand, they can also alleviate the problem of insufficient legal talents and enable more non-legal professionals to obtain professional legal advice. ***Legal Judgment Prediction (LJP)***, which aims to infer the corresponding accusations based on given case descriptions, is a vital task in the field of Legal AI (Yuan et al., 2019; Feng et al., 2022). Fig. 1 illustrates the process of LJP, where a case description text is shown on the left, the candidate labels are shown on the right, and the label corresponding to the current case description text inferred by an AI algorithm is shown in the middle.

The LJP task is usually considered a classification problem. Most early LJP methods are rule-based or manual feature-based machine learning methods (Segal, 1984, Li et al., 2018). Rule-based methods are highly interpretable but cannot cope with diverse text forms. Manual feature-based machine learning methods usually achieve good results but rely on high-quality feature engineering. In recent years, with the development of deep learning techniques in the natural language processing community, many neural network models for LJP have been proposed (Chalkidis et al., 2019; Yang et al., 2019; Zhong, Wang et al., 2020; Lyu et al., 2022). For example, (Sukanya and Priyadarshini, 2021) proposed an attention-based model, (Chen et al., 2019) proposed a gating network-based model, and (Yang et al., 2019) proposed a recurrent neural network-based model. However, these methods invariably require feeding the neural networks with a large amount of labeled data.

In addition to designing novel neural network models, there is a simple way to implement LJP by fine-tuning a Pre-trained Language Model (PLM) on the task data. Since BERT was proposed in 2019, fine-tuning PLMs using task-specific labeled data has been successful on

[*] Corresponding author.
E-mail addresses: sunjingyun@hrbeu.edu.cn (J. Sun), huangshaobin04@126.com (S. Huang), weichi2022@126.com (C. Wei).

https://doi.org/10.1016/j.eswa.2023.122177
Received 28 January 2023; Received in revised form 28 June 2023; Accepted 13 October 2023
Available online 20 October 2023

many tasks and has become a paradigm of natural language processing today (Clark et al., 2021, Li et al., 2021; Loukas et al., 2022). The power of PLMs is mainly due to the large amount of general semantic knowledge learned during their pre-training on the large-scale corpus (Roberts et al., 2020, Talmor et al., 2020). In other words, downstream tasks are to load model checkpoints that have been pre-trained instead of randomly initializing the models' parameters. Until today, this method has been a strong baseline for many tasks. However, large-scale labeled data is still required to fine-tune a PLM adequately. In realistic scenarios, obtaining labeled data in the legal field is usually expensive. Therefore, designing an LJP method that remains effective in the few-shot scenario is necessary.

Prompt learning is a recently popular methodology that performs exceedingly in few-shot and even zero-shot scenarios (Gao et al., 2021, Schick and Schütze 2021, Zhu et al., 2022). The core idea of prompt learning is to convert a classification task into a masked language model task using a prompt template. This enables downstream tasks to be aligned with PLMs' pre-training task, thus stimulating the PLMs' recall of the semantic knowledge learned. In addition, this guides PLMs to understand the downstream tasks through the template content. Prompt learning has already achieved remarkable results on many tasks, such as text classification (Han et al., 2021), text entailment (Schick and Schütze, 2021), and entity linking (Ding et al., 2021; Zhu et al., 2022). Therefore, it is reasonable to believe that prompt learning can provide an alternative solution for LJP tasks in the few-shot scenario.

In this work, we propose a knowledgeable prompt learning-based method called ***KnowPrompt4LJP*** for the Chinese LJP task. We first convert LJP from a classification task to a masked language model task employing an elaborate prompt template. This can stimulate the PLM to recall the knowledge learned in its pre-training and enable the PLM to understand the Chinese LJP task better. Moreover, it has been shown that keyword information in Chinese case descriptions benefit the task. Therefore, we match the keywords in Chinese case descriptions via an external knowledge base and then incorporate the keyword information into the soft prompt tokens in the prompt template to enhance the template's guidance for the PLM. Finally, there has always been a severe problem with the Chinese LJP task. Most mainstream PLMs are Transformer-based; limited by computational complexity, the input length of these PLMs is limited to 512 tokens. However, realistic Chinese case descriptions are often very long. In the CAIL2018 competition dataset, there are more than 200,000 case descriptions longer than 512 tokens (Xiao et al., 2018). Therefore, the current mainstream PLMs, such as BERT, RoBERTa, DeBERTa, etc., cannot be directly applied to the Chinese LJP task. We use a recently open-sourced PLM checkpoint called Lawformer (Xiao et al., 2021), pre-trained on a large-scale Chinese legal corpus using the LongFormer model, to solve this problem. Unlike the traditional Transformer, LongFormer introduces dilated attention and sliding attention, thus allowing text input of more than 1,000 tokens. Also, since Lawformer is pre-trained on a Chinese legal corpus, it is more suitable for the task in the Chinese legal field.

To evaluate the feasibility and effectiveness of our KnowPrompt4LJP, we conducted extensive experiments on the ***CAIL2018*** dataset. To the best of our knowledge, CAIL2018 is the best quality and largest Chinese LJP dataset to date. Firstly, we conducted zero-shot and few-shot experiments on the dataset. The experimental results show that KnowPrompt4LJP far outperforms existing LJP methods in both zero-shot and few-shot scenarios. In particular, the Macro F1 value achieved by KnowPrompt4LJP improves by 0.3 over that achieved by the standard fine-tuning BERT method in the few-shot setting. In addition, we evaluate the performance of KnowPrompt4LJP on the full-scale training data. The experimental results show that KnowPrompt4LJP also outperforms all the baselines under the condition of using the whole training data. Compared to standard fine-tuning BERT, KnowPrompt4LJP improves the Macro F1 value by 0.20 to 0.81.

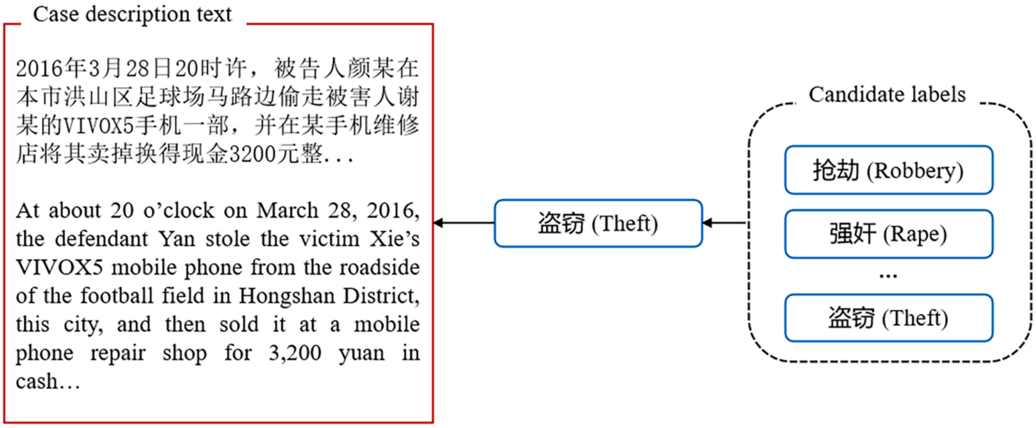
Our main contributions can be summarized as follows.

- In this work, we propose a novel Chinese legal judgment prediction method based on the prompt learning framework.
- We incorporate keyword information extracted via a knowledge base into soft prompt tokens to enhance the prompt template's guidance for the PLM.
- We use Lawformer, a new checkpoint pre-trained on a large-scale Chinese legal corpus using the LongFormer model, as the PLM, thus allowing the input texts longer than 512 tokens.
- We conduct extensive experiments on a high-quality dataset and verified that our proposed method outperforms the baselines in both low-resource and data-rich scenarios.

The remainder of this paper is organized as follows. Section 2 presents related works, including research advances in LJP and the basic concepts and applications of prompt learning. Section 3 formally describes our task. Section 4 first introduces the overall structure of our KnowPrompt4LJP and then details each component. Section 5 describes the experimental setup. Section 6 presents the experimental results and provides a detailed analysis. Section 7 discusses the potential impact of our work. Section 8 concludes our work and looks forward to future work.

2. Related work

This section describes the works related to our study. Firstly, Section 2.1 introduces the research progress in LJP and summarizes the strengths and weaknesses of the existing methods. Then, Section 2.2 introduces the basic concepts and main applications of prompt learning



2016年3月28日20时许，被告人颜某在本市洪山区足球场马路边偷走被害人谢某的VIVOX5手机一部，并在某手机维修店将其卖掉换得现金3200元整...

At about 20 o'clock on March 28, 2016, the defendant Yan stole the victim Xie's VIVOX5 mobile phone from the roadside of the football field in Hongshan District, this city, and then sold it at a mobile phone repair shop for 3,200 yuan in cash…

Fig. 1. Example of the legal judgment prediction task.

and discusses the problems faced in applying it to the LJP task.

2.1. Legal judgment prediction

Legal Judgment Prediction (LJP) aims to return predicted accusations based on the case description texts entered by users. From a legal practitioner's perspective, LJP can provide decision support and thus improve productivity. From the standpoint of non-legal professionals, LJP can help them get professional legal advice anytime and anywhere. Therefore, LJP has always been an essential task in legal AI. With the rapid development of natural language processing technology, LJP has attracted wider attention from the natural language processing community in recent years.

Early LJP methods were mainly rule-based methods or handcrafted feature-based machine learning methods (Nagel, 1963, Lauderdale and Clark, 2012, Barros et al., 2018). Rule-based methods help to interpret the results and thus increase trust in the results. However, such methods cannot cope with diverse text forms and, therefore, hardly stand the test of sizeable unseen test instances. In contrast, manual feature-based machine learning methods have good generalization; they can obtain correct predictions based on the learned feature distributions even when some unseen test instances are encountered. For example, (LiuC, 2004) first obtained syntactic and semantic features in case description texts based on the keywords and then used KNN as a classifier to predict labels. This method achieved satisfactory results on a small-scale Chinese dataset. However, such methods still need to be improved in fitting data distribution because they can only learn some shallow manual features of the data. Therefore, such methods remain fragile in real scenarios.

With the development of deep learning techniques, most recent LJP methods are centered on elaborate neural network models. For example, (Wang et al., 2019) proposed a model based on convolutional neural networks for LJP. Unlike shallow machine learning classifiers, neural network models can automatically extract deep semantic features from case description texts. Moreover, the translational invariance of convolutional neural networks enables the model to focus on those valuable keywords or terms in case description texts, which results in more accurate predictions. In addition, (Yang et al., 2019) proposed a LJP model based on recurrent neural networks. Unlike convolutional neural networks, recurrent neural networks are better at modeling the global semantic dependencies of a case description text. However, the high computational complexity of recurrent neural networks leads to slower training and inference speed of the model. To guarantee training and inference speed, (Chen et al., 2019) proposed a gated network-based model for LJP. A gated network is a variant of a recurrent network with the same modeling ability as the recurrent network but with lower computational complexity. With the widespread use of attention mechanisms, (Sukanya and Priyadarshini, 2021) proposed an attention-based model for LJP and achieved satisfactory results on a larger-scale dataset.

In addition to elaborating novel neural network model structures for LJP tasks, there is a concise and practical LJP approach based on deep learning techniques, fine-tuning a BERT on the LJP training data. Since BERT was proposed in 2019, such method has achieved SOTA results on numerous natural language processing tasks (Clark et al., 2021, Li et al., 2021, Loukas et al., 2022). However, there is a severe problem in directly applying BERT to LJP tasks. The maximum length of input text allowed by BERT is 512 tokens, but most of the case description texts in LJP tasks are longer than 512 tokens. Some case description texts are even longer than 2,000 tokens. To use BERT effectively on LJP tasks, some studies have proposed a model structure based on hierarchical BERT. For example, (Chalkidis et al., 2019) first use BERT to read the tokens of each fact fragment to obtain fact-level embeddings. Then, they use Transformer to read the fact embeddings to get final case-level embeddings. Thanks to the large amount of general semantic knowledge learned by BERT during its pre-training, such methods usually achieve desirable results and do not require the effort of designing complex task-specific model structures.

Although the existing deep learning-based LJP methods have achieved remarkable results, they are all data-intensive methods. However, data labeling is costly in the legal AI field. Therefore, proposing a learnable model that relies on a manageable amount of training data is necessary. To this end, we offer an LJP method based on the prompt learning framework called KnowPrompt4LJP. Besides, under the prompt learning framework, we utilize the keyword information of case description texts to enhance KnowPrompt4LJP further.

2.2. Prompt learning

Prompt learning is a methodology that has become very popular recently. Unlike the standard approach of fine-tuning a PLM to adapt the PLM to downstream tasks, prompt learning is about adapting downstream tasks to a PLM. The core idea is to transform different downstream tasks into a masked language model task uniformly. On the one hand, this aligns the downstream tasks to the PLM's pre-training task, thus facilitating the PLM's recall of the semantic knowledge learned from pre-training. On the other hand, some prompt tokens related to the downstream tasks can be constructed in prompt templates, thus enhancing the PLM's understanding of the downstream tasks.

A typical prompt learning framework consists of three components: a prompt template, a PLM, and a label mapping. A ***prompt template*** is designed to wrap the text input of a downstream task into the input form of a masked language model task. For example, in the sentiment analysis task, the input text "*The film was badly made.*" can be wrapped by a prompt template as "*The film was badly made. It was [MASK]*" (Ding et al., 2022). Where "*It was*" are the prompt tokens and "*[MASK]*" is the masked token, i.e., the token to be predicted. In most prompt learning works, the prompt tokens in templates are designed manually; these are called hard or manual templates. In addition, templates in some works contain soft prompt tokens (Lester et al., 2021, Qin and Eisner, 2021). A ***soft prompt*** token is a type of token that its embedding vector is learnable. Adding soft prompt tokens to a template allows the model to automatically search for optimal prompts, thus further improving the model's effectiveness. A ***PLM*** is the engine of prompt learning for predicting tokens at masked locations. Despite the importance of a prompt template and label mapping, it has to be acknowledged that the final effect of prompt learning is also inextricably linked to the capability of a PLM. The commonly used PLMs are BERT, RoBERTa, AlBERTa, GPT, and T5. In addition, a domain-appropriate PLM is usually chosen for some downstream tasks in particular fields. For example, (Zhu et al., 2022) used ClinicBERT as a PLM in the medical entity linking task and achieved satisfactory results. ***Label mapping***, also called ***verbalizer*** in prompt learning, is used to map the results on the masked language model task back to downstream tasks. This process establishes a mapping between predicted words and task labels through label words. For example, in work of (Schick and Schütze, 2021) on textual entailment, there are three label words: "*yes*", "*no*", and "*maybe*". If the token predicted on the masked position is "*yes*", the relationship between the two sentences is entailment, if the token predicted on the masked position is "*no*", the relationship between the two sentences is contradiction; and if the token predicted on the masked position is "*maybe*", then the relationship between the two sentences is neutral. In some works, multiple label words corresponding to one label are also designed. For example, in the work of (Hu et al., 2022) on news classification, the category label of news is policy when the predicted word on the masked position is any one of "*government*", "*diplomatic*", and "*law*".

Prompt learning has achieved outstanding results on many downstream tasks in the few-shot scenario. For example, (Schick and Schütze, 2021) proposed a few-shot text classification method based on prompt learning, called iPET. The method can achieve more than 60 % accuracy on large-scale text classification datasets including Yelp, AG's, and Yahoo using only 50 training instances. Also, even using only 10 training instances, iPET can achieve up to 89 % accuracy on AG's and 71 % on

Yahoo. In addition, prompt learning has made many advances in information extraction tasks. For example, (Chen et al., 2021) proposed a relationship extraction method based on prompt learning, (Shin et al., 2021) proposed a semantic extraction method based on prompt learning, and a named entity recognition method based on prompt learning was proposed by (Ding et al., 2021). Therefore, we believe that prompt learning can provide an option for the LJP task in the few-shot or even zero-shot scenario.

In previous studies in prompt learning, label mapping is usually established between a single English token and a unique task label. In other words, there is only one masked token in a prompt template, and we only need to get the prediction on that masked token to get the corresponding category label. However, in our LJP task, many accusations cannot be described by a single Chinese token. Therefore, we need to set multiple masked tokens in a prompt template and propose a new mapping rule.

3. Task formulation

This work aims to predict the corresponding legal judgment based on the textual description of a case, which is essentially a classification problem. Formally, given the text $x = [t_1, t_2, \cdots t_N]$ of a case containing N tokens, the goal of the LJP task is to assign a label $y \in \{0, 1, \cdots K\}$ to it, where 0 to K represent the ids of the accusation labels.

Previous deep learning-based LJP approaches typically use the whole training data $\mathscr{D}_{train} = \{(x_1, y_1), (x_2, y_2), \cdots, \left(x_{|\mathscr{D}_{train}|}, y_{|\mathscr{D}_{train}|}\right)\}$ to train a neural network model from scratch or fine-tune a large PLM, where y_i denotes the ground truth label of the i_{th} training instance. However, in this work, we manage to use only a tiny amount of training data $\mathscr{D}_{few} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_{|\mathscr{D}_{few}|}, y_{|\mathscr{D}_{few}|})\}$ to fine-tune a large PLM and yet still make the PLM work well for the LJP task. In addition, we hope our method can also achieve SOTA results when using the whole training data.

4. Method

Our proposed KnowPrompt4LJP consists of four components: 1) Prompt template; 2) Knowledge prompt; 3) Label mapping; and 4) Lawformer, as shown in Fig. 2. ***Prompt template*** wraps the original case description text x into x' that matches the masked language model task, and the template's content can enhance the PLM's understanding of the LJP task. ***Knowledge prompt*** matches keyword information from the original input x via an external knowledge base and injects the information into the soft prompt tokens in x'. ***Label mapping*** maps the results obtained from the masked language model task back to the classification task's results. In other words, the label mapping assigns the final category label $\hat{y} \in \{0, 1, 2, \cdots, K\}$ to the case description x based on the predicted tokens at the masked positions. 4) ***Lawformer*** is the PLM checkpoint used in our KnowPrompt4LJP, which is pre-trained on a large-scale Chinese legal corpus using the LongFormer model. Lawformer allows text input of more than 512 tokens and thus can model richer information about case descriptions.

Next, we will detail the four components of KnowPrompt4LJP: Section 4.1 introduces the prompt template, Section 4.2 introduces the knowledge prompt, Section 4.3 introduces the label mapping, and Section 4.4 introduces Lawformer, the PLM we used in KnowPrompt4LJP.

4.1. Prompt template

This section describes the prompt template in our KnowPrompt4LJP. A prompt template is one of the essential components of prompt learning, which is used to wrap original textual input into the input form of masked language model tasks. Besides, the tokens in a template can also provide information to enable the PLM to understand a downstream task. Therefore, templates containing different tokens for various downstream tasks need to be designed. We consider that 1) the essence of the LJP task is to assign an accusation to a defendant based on the case description, and 2) the cases in CAIL2018 dataset, the dataset used in this work, are all criminal cases. Therefore, we design the prompt template as shown below:

“以[M][M][M][M][M][M][M][M][M][M]罪对其进行刑事判决:*x*”.

Its English version is:

“He will be held criminally responsible for the crime of [M] [M] [M] [M] [M] [M] [M] [M] [M] [M]: *x*”.

where “*[M]*” denotes the token [MASK], i.e., the token to be predicted, and x is the original case description text. In most existing prompt learning efforts, a prompt template contains only one masked

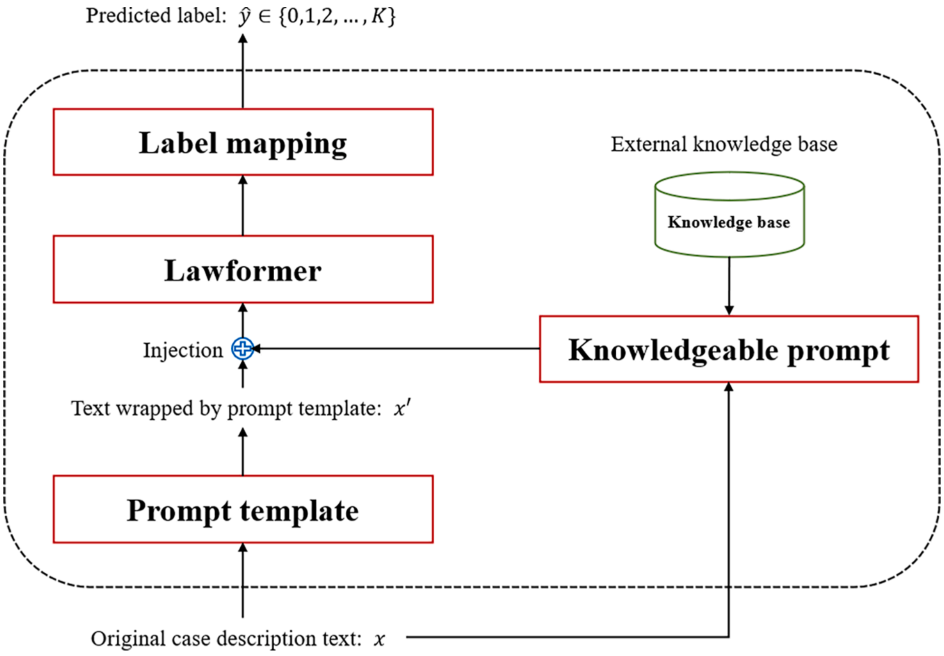

Predicted label: $\hat{y} \in \{0,1,2,\ldots,K\}$
Label mapping
External knowledge base
Knowledge base
Lawformer
Injection
Knowledgeable prompt
Text wrapped by prompt template: x'
Prompt template
Original case description text: x

Fig. 2. Structure diagram of our proposed KnowPrompt4LJP.

token. For example, in the work of (Hu et al., 2022), the prompt template is "*This is a [M] question: x*" which contains only one masked token. Besides, in the work of (Ding et al., 2021) on entity linking, the prompt template is constructed as "*x. e_x is a [M]*", which also contains only one masked token, where e_x is the name of the entity contained in x. In contrast, the template we constructed contains up to ten masked tokens. This is because, in the Chinese LJP task, only using one Chinese token cannot describe a complete accusation. For example, we cannot use only one Chinese token to describe accusation labels "抢劫(Robbery)", "入室盗窃(Burglary)", and "破坏公共设施(Destruction of public facilities)". We consider that all the accusation labels in the CAIL2018 dataset can be described well using less than ten Chinese tokens and the experiments in Section 5.3 verified this.

Given the prompt template *Template*$(\cdot)$, KnowPrompt4LJP wraps the original case description text x as x', the input form of mask language model tasks, as shown in formula (1).

$$x' = \left[t_{T_1}, m_1, m_2, \cdots, m_{10}, t_{T_2}, t_{T_3}, \cdots t_{T_M}, t_1, t_2, \cdots, t_N\right] = Template(x = [t_1, t_2, \cdots, t_N]) \tag{1}$$

where t_{T_i} denotes the i_{th} prompt token in the template, m_i denotes the i_{th} masked token in the template, and t_i indicates the i_{th} token in the original case description text.

4.2. Knowledgeable prompt

Previous researches have shown that keyword information contained in case description texts helps the LJP task (Hu et al., 2018, Zhong et al., 2018). Therefore, we first extract keywords from the case description texts via an external knowledge base. Further, we incorporate the extracted keyword information into the prompt template to enhance the template's guidance for the PLM. These two processes are described below.

We use ***THUOCL_Law*** as the external knowledge base. THUOCL_Law is a subbase of the Tsinghua University Open Chinese Lexicon (THUOCL), a high-quality Chinese lexicon compiled and launched by the Natural Language Processing and Social Humanities Computing Laboratory of Tsinghua University, in which all subbases have undergone multiple rounds of manual screening to ensure the accuracy. Table 1 shows some of the words in THUOCL_Law. We extract keywords from the case description texts by simple rule matching. With the external knowledge base THUOCL_Law, the keywords extracted from the original case description text x form the set $\mathscr{A} = \{a_1, a_2, \cdots a_{|\mathscr{A}|}\}$, where a_i denotes the i_{th} keyword extracted.

To incorporate keyword information into the prompt template, we first introduce the concept of soft prompt tokens. The concept of soft prompt tokens was first introduced in the work of (Li and Liang, 2021). Its core idea is to automatically search for the optimal prompts for a PLM by learning continuous vectors of prompt tokens. Specifically, some tokens called soft prompts are inserted into a prompt template. Unlike human-readable token, the embedding vectors of these soft prompt tokens are learnable and can therefore be optimized during the training. To illustrate the process of knowledge incorporation, we represent the PLM in two parts: *PLMEmbedding*$(\cdot)$ denotes the embedding layer of the PLM, and *PLMEncoder*$(\cdot)$ denotes the encoder of the PLM. In a PLM, the embedding layer is used to embed tokens into corresponding dense vectors, while the encoder performs feedforward computation on the embedded vectors to obtain hidden layer outputs, i.e., the contextual representations of the input tokens. We first inserted two soft prompt tokens into the original prompt template to obtain a new prompt template:

"[S]以[M][M][M][M][M][M][M][M][M][M]罪对其进行刑事判决: [S]*x*".

Its English version is:

"[S] He will be charged with criminal responsibility for [M] [M] [M] [M] [M] [M] [M] [M] [M] [M]: [S] *x*".

Where [S] denotes a soft prompt token whose embedding vector is learnable. Thereby, given the original case description x, the prompt template wraps it into x' containing soft prompt tokens, and the process is shown in formula (2).

$$x' = \left[s_1, t_{T_1}, m_1, m_2, \cdots, m_{10}, t_{T_2}, t_{T_3}, \cdots t_{T_M}, s_2, t_1, t_2, \cdots, t_N\right] = Template(x = [t_1, t_2, \cdots, t_N]) \tag{2}$$

where s_i denotes the i_{th} soft prompt token in the prompt template. Formula (2) differs from formula (1) by adding two soft prompt tokens, s_1 and s_2. Fig. 3 shows the wrapping of the prompt template visually through an example. An original case description is shown at the bottom of the figure, and a masked language model input wrapped by our knowledgeable prompt template is shown at the top.

Next, all the tokens in x' except the soft prompt tokens are embedded by the embedding layer of the PLM. At the same time, the soft prompt tokens in x' are embedded by an additional trainable embedding matrix. The process is shown in formula (3).

$$\boldsymbol{e}_i = \begin{cases} P[i], ifi \in soft_{idx} \\ PLMEmbedding(token_i), otherwish \end{cases} \tag{3}$$

where $\boldsymbol{P} \in \mathbb{R}^{|soft_{idx}| \times d_h}$ is the trainable embedding matrix, and $soft_{idx}$ is the index of the soft prompt tokens. d_h is the hidden layer dimension of the PLM. The embedding vectors $\boldsymbol{E}$ of all the tokens (including the soft prompt tokens) in x' can be obtained by the formula.

$$\boldsymbol{E} = \left[\boldsymbol{e}_{s_1}, \boldsymbol{e}_{T_1}, \boldsymbol{e}_{m_1}, \boldsymbol{e}_{m_2}, \cdots, \boldsymbol{e}_{m_{10}}, \boldsymbol{e}_{T_2}, \boldsymbol{e}_{T_3}, \cdots, \boldsymbol{e}_{T_M}, \boldsymbol{e}_{s_2}, \boldsymbol{e}_{t_1}, \boldsymbol{e}_{t_2}, \cdots, \boldsymbol{e}_{t_N}\right]$$

where $\boldsymbol{e}_{s_i}$ denotes the embedding vector of the i_{th} soft prompt token, $\boldsymbol{e}_{T_i}$ denotes the embedding vector of the i_{th} regular prompt token, $\boldsymbol{e}_{m_i}$ denotes the embedding vector of the i_{th} masked token, and $\boldsymbol{e}_{t_i}$ denotes the embedding vector of the i_{th} token in the original case description x.

Meanwhile, we concatenate all the keywords in the set $\mathscr{A}$ into a text fragment and model the text fragment using the BiGRU proposed by (Chung et al., 2014). The process is shown in formula (4).

$$\boldsymbol{a} = BiGRU\left(\left[a_1, a_2, \cdots, a_{|\mathscr{A}|}\right]\right) \tag{4}$$

The dense vector $\boldsymbol{a}$ now contains the keyword information of the case description x. Since BiGRU is a commonly used neural network structure, we do not detail it here.

Next, we inject the dense vector $\boldsymbol{a}$ containing keyword information into the embedding vectors $\boldsymbol{e}_{s_1}$ and $\boldsymbol{e}_{s_2}$ of the soft prompt tokens to obtain two new vectors. The process is shown in formula (5) and (6).

$$\widetilde{\boldsymbol{e}}_{s_1} = \boldsymbol{e}_{s_1} + \boldsymbol{a} \tag{5}$$

$$\widetilde{\boldsymbol{e}}_{s_2} = \boldsymbol{e}_{s_2} + \boldsymbol{a} \tag{6}$$

Then, we replace $\boldsymbol{e}_{s_1}$ and $\boldsymbol{e}_{s_2}$ in $\boldsymbol{E}$ using $\widetilde{\boldsymbol{e}}_{s_1}$ and $\widetilde{\boldsymbol{e}}_{s_2}$ to obtain:

$$\widetilde{\boldsymbol{E}} = \left[\widetilde{\boldsymbol{e}}_{s_1}, \boldsymbol{e}_{T_1}, \boldsymbol{e}_{m_1}, \boldsymbol{e}_{m_2}, \cdots, \boldsymbol{e}_{m_{10}}, \boldsymbol{e}_{T_2}, \boldsymbol{e}_{T_3}, \cdots, \boldsymbol{e}_{T_M}, \widetilde{\boldsymbol{e}}_{s_2}, \boldsymbol{e}_{t_1}, \boldsymbol{e}_{t_2}, \cdots, \boldsymbol{e}_{t_N}\right]$$

Finally, the encoder of the PLM takes $\widetilde{\boldsymbol{E}}$ as input and obtains hidden layer outputs, i.e., the contextual representations of the input tokens, through multilayer feedforward operations. The process is shown in formula (7).

$$\boldsymbol{H} = \left[\boldsymbol{h}_{s_1}, \boldsymbol{h}_{T_1}, \boldsymbol{h}_{m_1}, \boldsymbol{h}_{m_2}, \cdots, \boldsymbol{h}_{m_{10}}, \boldsymbol{h}_{T_2}, \boldsymbol{h}_{T_3}, \cdots, \boldsymbol{h}_{T_M}, \boldsymbol{h}_{s_2}, \boldsymbol{h}_{t_1}, \boldsymbol{h}_{t_2}, \cdots, \boldsymbol{h}_{t_N}\right] = PLMEncoder\left(\widetilde{\boldsymbol{E}}\right) \tag{7}$$

Table 1
Part of the words in THUOCL_Law.

违背妇女意志(against women's will), 违约(breach of contract), 拐卖(kidnapping), 抢夺(snatch), 殴打(beat up), 致残(disabled), 故意(deliberately), 残忍(cruel)

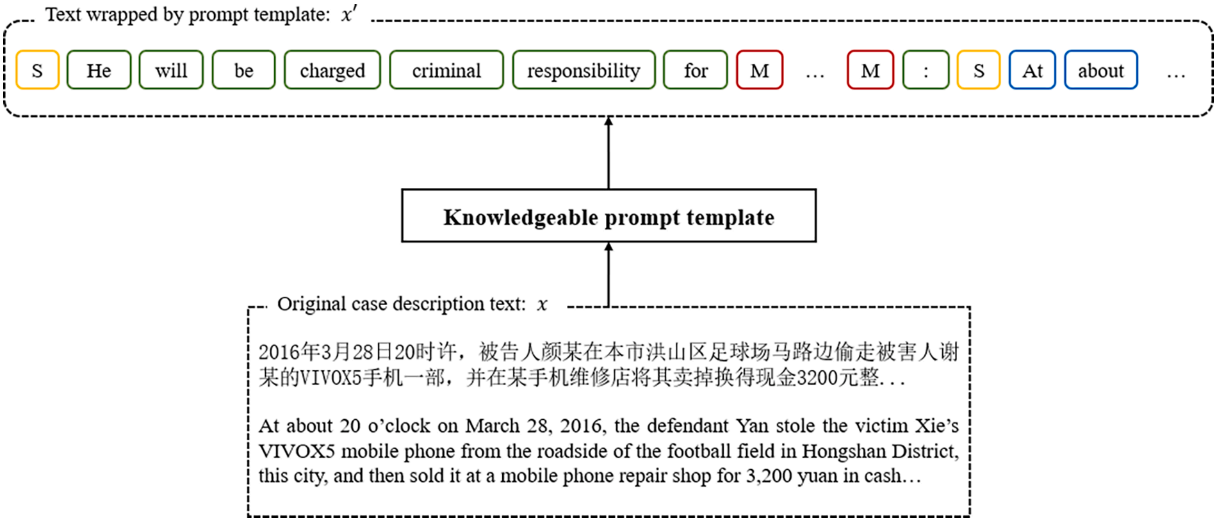

Text wrapped by prompt template: x'
S
He
will
be
charged
criminal
responsibility
for
M
…
M
:
S
At
about
…
Knowledgeable prompt template
Original case description text: x
2016年3月28日20时许，被告人颜某在本市洪山区足球场马路边偷走被害人谢某的VIVOX5手机一部，并在某手机维修店将其卖掉换得现金3200元整...
At about 20 o'clock on March 28, 2016, the defendant Yan stole the victim Xie's VIVOX5 mobile phone from the roadside of the football field in Hongshan District, this city, and then sold it at a mobile phone repair shop for 3,200 yuan in cash…

Fig. 3. Example of our prompt template wrapping an original case description text.

4.3. Label mapping

Label mapping is also an essential component in prompt learning, which aims to map the results of masked language model tasks back to the results of classification tasks. Our label mapping differs from the label mappings in the existing works. Label mappings in most existing works map the predicted result on a single masked token to a unique category label. In other words, their prompt templates contain only one masked token. Their label mappings only need to assign final category label based on the prediction on one masked token. However, in our work, the prompt template contains up to ten masked tokens. Therefore, we need to assign the final category label based on the overall results of ten masked tokens rather than considering only one masked token. To this end, we first construct label words containing multiple tokens for each category label. Then, we determine the final labels through the similarity between the predicted results and the label words. These two processes are detailed below.

In this work, we directly use accusation texts themselves as the label words of the corresponding category label. For example, the category label $0 \in \{0, 1, 2, \cdots, K\}$ corresponds to the endanger public safety accusation; we directly use the text "*Endanger public safety*" as the label word corresponding to the label 0. In the CAIL2018 dataset, there are 196 different accusations. Therefore, we finally get 196 corresponding label words, denoted as $\mathscr{V} = \{v_0, v_1, \cdots, v_K\}$, as a bridge to connect the masked language model task with the classification task. Table 2 shows some of our label words.

However, using accusation texts directly as label words raises a tricky problem: in the LJP task, different accusations contain different numbers of tokens, yet the number of masked tokens contained in our prompt template is fixed. To solve the problem, all label words with less than ten tokens are padded by token [PAD], while label words with more than ten tokens are truncated. In this way, all label words contain ten tokens, equal to the number of masked tokens in our prompt template. We use Jaccard similarity to measure the similarities between the predicted results and the label words to determine the final category labels. The process is shown in formula (8).

$$\hat{y} = \underset{y\in\{0,1,\cdots K\}}{argmax}\left(Jaccard\left(\hat{m}_{1:10}, v_y\right)\right) \tag{8}$$

where $\hat{y}$ is the final label id assigned by KnowPrompt4LJP to the case description text x, and $\hat{m}_{1:10}$ denotes the continuous fragment consisting of the predicted tokens. Note that token [PAD] is not considered in the calculation of Jaccard similarity. To make readers understand the label mapping of KnowPrompt4LJP intuitively, we describe the above process through an example shown in Fig. 4.

Table 2
Some of our label words.

Label word	制造、贩卖、传播淫秽物品(Manufacture, sell, and disseminate obscene materials) 非法持有、私藏枪支、弹药(Illegal possession of firearms and ammunition) 非法占用农用地(Illegal occupation of agricultural land) 非法种植毒品原植物(Illegal cultivation of narcotic plants) 危害公共安全(Endanger public safety)

4.4. Lawformer

In addition to the prompt template and label mapping, a PLM also plays an important role in prompt learning. A PLM takes the texts in the form of masked language model tasks as input and predicts masked tokens. It has been shown that choosing an appropriate PLM helps prompt learning models to get better performances (Liu et al., 2021).

Compared to common text classification tasks, case description texts in the LJP task are extremely long. In the CAIL2018 dataset, more than 200,000 cases are longer than 512 tokens. Therefore, some recently popular PLMs, such as BERT, RoBERTa, DeBERTa, etc., cannot directly apply to these case description texts. To address this problem, we use Lawformer as our PLM checkpoint. Lawformer is a checkpoint pre-trained on a large-scale Chinese legal corpus using the LongFormer model by (Xiao et al., 2021). The LongFormer model is a variant of the Transformer model that reduces computational complexity by introducing sliding attention and dilated sliding attention to allow for inputs of well over 512 tokens. Fig. 5 shows in a visual way how sliding attention and dilated sliding attention reduce the computational complexity compared to Transformer's standard attention. The shades of color of the small squares in the figure represent the attention relationship between the tokens in the text.

In addition, Lawformer is naturally suitable for the LJP task because its training corpus is based on legal texts. This means that Lawformer's semantic space is closer to the semantic space of the case description texts than that of regular BERT, RoBERTa, and DeBERTa and thus can fit data distribution better and faster. In this work, Lawformer is only used as an out-of-the-box PLM checkpoint, so we do not detail its model structure and pre-training algorithm.

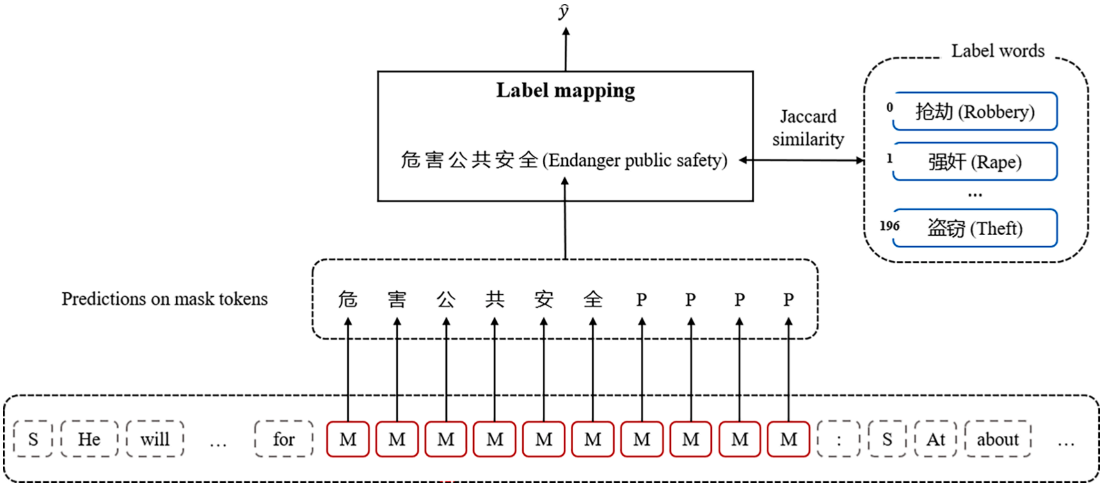
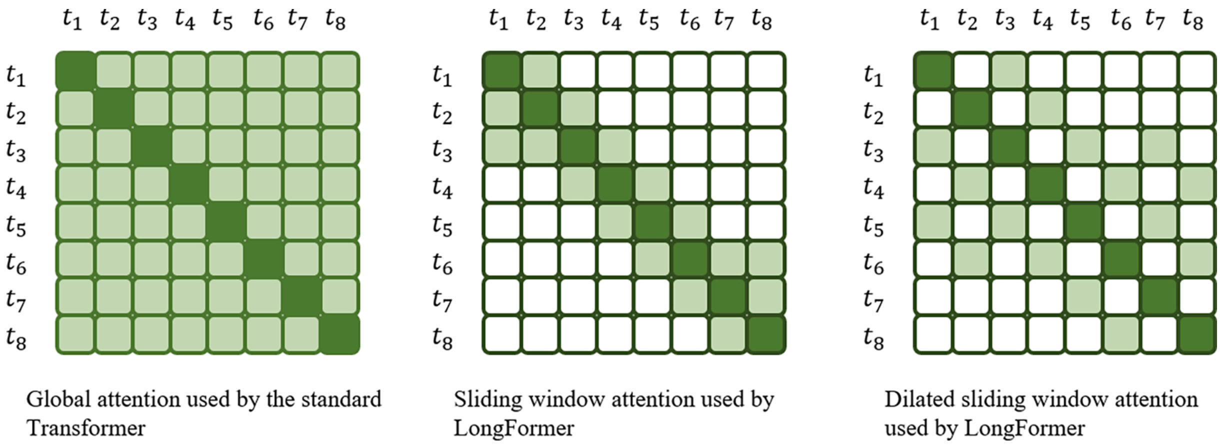

ŷ
Label mapping
危害公共安全 (Endanger public safety)
Jaccard similarity
Label words
0 抢劫 (Robbery)
1 强奸 (Rape)
...
196 盗窃 (Theft)
Predictions on mask tokens
危 害 公 共 安 全 P P P P
S He will … for M M M M M M M M M M : S At about …

Fig. 4. An example of label mapping.



Global attention used by the standard Transformer

Sliding window attention used by LongFormer

Dilated sliding window attention used by LongFormer

Fig. 5. Compared with the standard Transformer structure, LongFormer reduces computational complexity by introducing sliding window attention and dilated sliding window attention.

5. Experiment settings

In this section, we first introduce CAIL2018, the dataset used for our experiments. Then, we present implementation details, including the settings of hyperparameters. Finally, we introduce the baselines used for the comparison experiments.

5.1. Dataset

The experimental dataset we use is the CAIL2018 competition dataset (Xiao et al., 2018), which is the largest Chinese LJP task dataset with high annotation quality to date. The authors acquired over 2.6 million criminal cases published by the Supreme People's Court of China on https://wenshu.court.gov.cn/. After pre-processing, the dataset finally contains 2,676,075 case description texts and 196 unique accusations. Each case description text corresponds to only one accusation label, so the task is a single-label classification problem. Table 3 shows some instances from the CAIL2018 dataset. In addition, 2/3 of the total are used as the training set and the remaining 1/3 as the test set.

Additional statistical information on the CAIL2018 dataset is shown in Table 4. As we can see from the first row of the table, some of the case description texts even contain more than 5,000 tokens, which is far beyond the input limit of regular PLMs such as BERT. Also, from the fourth row of the table, we can see that the most extended accusation text contains 32 tokens. Finally, we can see from the last row of the table

Table 3
Some instances in the CAIL2018 dataset.

Case description text	Accusation label
被告人罗某甲…, 罗某甲踢了项某乙一脚, 之后双方发生互殴, … Defendant Luo…, Luo kicked Xiang, and then the two sides fought each other, …	故意伤害 Intentional injury
被告人黄某携带作案工具螺丝刀…, 后转售后得赃款… Defendant Huang carried a screwdriver as a crime tool…, and then resold it for money…	盗窃 Theft
被告人周某在本县武康街道营盘小区…窃得黑色苹果7PLUS手机一部… Defendant Zhou stole a black Apple 7PLUS mobile phone from Yingpan Community, Wukang Street…	盗窃 theft

Table 4
Statistics on the CAIL2018 dataset.

Maximum length of case description text	5,694
Minimum length of case description text	8
Average length of case description text	356
Maximum length of accusation label text	32
Minimum length of accusation label text	2
Average length of accusation label text	5
# Case description text longer than 512 tokens	241,434

that there are more than 200,000 case descriptions longer than 512 tokens.

5.2. Implementation details

We implement KnowPrompt4LJP based on the Pytorch and OpenPrompt libraries. Pytorch is the most popular deep learning implementation library, and OpenPrompt is a convenient prompt learning implementation library. We use the HuggingFace’s Transformers library to load Lawformer. Our model was trained and tested on an A100 GPU with 40G memory.

The optimizer of our model is AdamW, with a learning rate of 2e-5. The loss function used is the masked language model loss function. The word embeddings used in encoding keyword information are 200-dimensional embeddings trained via a Word2Vec model. In addition, the maximum length of the model input is set to 2,000; the texts longer than 2,000 tokens are truncated, and the texts shorter than 2,000 tokens are padded. The batch size is set to 4. When we experiment using the whole training set, the maximum epoch is set to 3; in the few-shot scenario, the maximum epoch is set to 15. For every 5,000 training steps, the loss on the development set is calculated. If the loss on the development set is no longer decreasing, the training is stopped early.

On the Chinese dataset, the backbone used in the BERT-based baselines is bert-base-chinese which can handle Chinese. On the English datasets, the backbone used in the BERT-based baselines is Legal-BERT pre-trained on legal corpus.

5.3. Baselines

To evaluate the effectiveness of our proposed KnowPrompt4LJP, we compare KnowPrompt4LJP with the following baselines. First, we introduce several recently proposed neural network-based LJP methods. Second, as described in Section 2.1, fine-tuning a PLM directly using the training data of the LJP task is an easy-to-implement but effective baseline. The baselines are described below.

(1) ***CNN***: We use *CNN* to represent the LJP method based on convolutional neural networks proposed by (Wang et al., 2019). The method uses convolutional neural networks to focus the model on key terminologies or key text fragments in case description texts.
(2) ***BiLSTM***: We use *BiLSTM* to represent the LJP method based on bidirectional long short-term memory networks proposed by (Yang et al., 2019). The method models the global semantics of case description texts via bidirectional long and short-term memory networks.
(3) ***BiGRU***: We use *BiGRU* to represent the LJP approach based on bidirectional gated networks proposed by (Chen et al., 2019). Compared to bidirectional long and short-term memory networks, bidirectional gated networks have lower computational complexity for modeling the global semantics of case description texts.
(4) ***Attenion***: We use *Attention* to denote the attention mechanism-based LJP method proposed by (Sukanya and Priyadarshini, 2021). Attention mechanisms can assign different weights to factual information in different parts of a case description text. This is a SOTA method for the LJP task recently.
(5) ***BERT_FT***: We use *BERT_FT* to denote the method of fine-tuning a BERT on the LJP training data. It has been shown that this is a powerful baseline (Zhong, Xiao et al., 2020).
(6) ***HBERT***: We use *HBERT* to represent the LJP method based on hierarchical BERTs proposed by (Chalkidis et al., 2019). In this method, the whole case description text is divided into several fact segments, and then, each fact segment is read by BERT to obtain a fact-level representation. Finally, a Transformer model encodes the fact-level representations into a case-level representation. This is the SOTA method on an English LJP dataset.
(7) ***HMN***: HMN is a hierarchical matching network for Crime classification proposed by (Wang et al., 2019). This method is a novel and strong baseline on the CAIL2018 dataset.
(8) ***KnowPrompt4LJP***: This is our proposed LJP method based on knowledgeable prompt learning. The method enhances the PLM’s understanding of the LJP task through a prompt template. The method incorporates keyword information extracted from case description texts via an external knowledge base into the prompt template. See Section 4 for the details.

6. Experimental results and analysis

This section presents the experimental results and analyzes them. Section 6.1 presents the methods’ performance under different data scenarios and analyzes them. Section 6.2 analyzes the effectiveness of each component in KnowPrompt4LJP through ablation experiments. Section 6.3 further analyzes the performance of KnowPrompt4LJP through qualitative analysis.

6.1. Main results analysis

Table 5 shows the experimental results of different methods under different data scenarios. The evaluation metrics are macro precision, macro recall, and macro F1. The first column of the table indicates the number of shots. We sampled each different category label. For example, two instances are sampled for each label, totaling 392 instances sampled for 196 category labels. Our main reason for this sampling strategy is two problems exist in the CAIL2018 dataset. The first is that there are too many categories in the dataset, with a total of 196 various category labels; thereby, some category labels may be missed if we sample according to the total number. The second is that the dataset has long-tailed distribution characteristics, so sampling according to the total number may result in the category labels with a small instance amount being difficult to be sampled.

Table 5
Performance of the methods on the CAIL2018 dataset.

Shot/Per label	Method	MacroP	MacroR	MacroF1
0	*CNN*	0.06	0.03	0.04
	BiLSTM	0.04	0.03	0.03
	BiGRU	0.05	0.01	0.02
	Attention	0.03	0.03	0.03
	BERT_FT	0.13	0.11	0.12
	HBERT	0.14	0.08	0.10
	KnowPrompt4LJP (ours)	**0.35**	**0.34**	**0.34**
2	*CNN*	0.18	0.16	0.17
	BiLSTM	0.11	0.11	0.11
	BiGRU	0.12	0.10	0.11
	Attention	0.17	0.16	0.17
	BERT_FT	0.23	0.20	0.21
	HBERT	0.29	0.29	0.29
	KnowPrompt4LJP (ours)	**0.50**	**0.46**	**0.48**
8	*CNN*	0.44	0.44	0.44
	BiLSTM	0.32	0.31	0.32
	BiGRU	0.34	0.32	0.33
	Attention	0.47	0.46	0.47
	BERT_FT	0.39	0.40	0.40
	HBERT	0.51	0.50	0.50
	KnowPrompt4LJP (ours)	**0.72**	**0.69**	**0.70**
Full	*CNN*	0.73	0.72	0.73
	BiLSTM	0.69	0.67	0.68
	BiGRU	0.66	0.66	0.66
	Attention	0.77	0.76	0.76
	BERT_FT	0.62	0.61	0.61
	HBERT	0.77	0.78	0.77
	HMN	80.9	61.9	66.5
	KnowPrompt4LJP (ours)	**0.82**	**0.81**	**0.81**

Firstly, as seen from the table’s first part, all the baselines fail almost completely in the zero-shot scenario. This is a reasonable result because all of these baselines are neural network-based supervised learning methods that must be trained with sufficient labeled data. If the neural networks are not trained with labeled data, the output of these methods is equivalent to random. However, in contrast, we see that *KnowPrompt4LJP* still achieves a macro F1 value of 0.34 in the zero-shot scenario. This fully demonstrates the feasibility and effectiveness of our *KnowPrompt4LJP*. Although *KnowPrompt4LJP* is also essentially a neural network-based method, it has two main advantages over other baselines that enable it to achieve satisfactory results without training data. Firstly, *KnowPrompt4LJP* aligns the LJP task with the PLM's pre-training task, allowing the PLM to fully recall what it learned in pre-training. Secondly, *KnowPrompt4LJP* enables the PLM to understand the LJP task by the prompt template.

As seen from the table's second and third parts, the macro F1 values of *KnowPrompt4LJP* far exceed those of other baselines in the 2-shot and 8-shot settings. In particular, in the 8-shot scenario, the macro F1 value of *KnowPrompt4LJP* has reached 0.70, which is close to or even exceeds the results of some baselines using the whole training data. In addition, we see that *HBERT* achieves the second-best results in the 2-shot and 8-shot scenarios. The advantage of *HBERT* over other baselines is mainly in two aspects. On the one hand, BERT, the PLM it uses, has already learned a certain degree of general semantic knowledge in pre-training, so there is no need to train its model parameters from scratch. On the other hand, *HBERT* can model a complete case description text through hierarchical processing. However, we see that *HBERT* still has a huge gap of almost 0.2 in macro F1 values compared to KnowPrompt4LJP. This is because Lawformer, the pre-trained language model used in our KnowPrompt4LJP, can directly and accurately characterize a long case description text instead of obtaining a hierarchical representation of the text and therefore can obtain more coherent semantic information. Moreover, compared with the BERT checkpoint used in *HBERT*, the Lawformer used in our *KnowPrompt4LJP* is pre-trained on a Chinese legal corpus, and thus, it is more consistent with the semantic distribution of the case description texts in the LJP task. Besides, we also see that *CNN*, *BiLSTM*, *BiGRU*, *Attention*, and *BERT_FT* perform not very well in both 2-shot and 8-shot scenarios. This is because there are too few training instances fed to these supervised methods so that the neural networks cannot adequately fit the distribution of the data.

As seen from the last part of the table, our *KnowPrompt4LJP* can still achieve optimal results using the whole training data. We believe that introducing knowledge plays a key role now, in addition to the advantages of prompt learning itself. Sections 6.2 and 6.3 will further validate this idea. We observe that *BERT_FT*, which typically performs well on various tasks, performs mediocrely on the LJP task. We believe this is due to the presence of a large amount of case description texts with more than 512 tokens in the CAIL2018 dataset, which prevents us from feeding an entire text directly into the regular BERT to obtain an accurate representation of the text. We see that *CNN* outperforms *BiLSTM* and *BiGRU*. This is mainly because case description texts in the LJP task are very long; thus, recurrent neural networks lose many long-range semantic dependencies in modeling a text globally. In contrast, *CNN* does not focus on the global semantics of a text but only on the local key information in a text. *Attention* is more effective than *CNN*, *BiLSTM*, and *BiGRU*, mainly because the attention mechanism is better at modeling long-range semantic dependencies. Thus, it is easier to obtain complete global semantic information of a case description text.

Overall, our *KnowPrompt4LJP* can achieve the best results in zero-shot, few-shot, and full training data scenarios. This fully demonstrates the advantages of prompt learning on the LJP task and the ability of our designed prompt template and label mapping to guide PLM to implement the LJP task better. Also, it proves the effectiveness of incorporating keyword information from case description texts into the prompt template through an external knowledge base.

6.2. *Ablation analysis*

In this section, we further analyze the effectiveness of each component in KnowPrompt4LJP through ablation experiments on the CAIL2018 dataset. Firstly, we remove the knowledgeable prompt. As can be seen in Table 6, this causes our KnowPrompt4LJP's macro F1 values to drop significantly in both shot-8 and full-size training data scenarios. In particular, the macro F1 value of KnowPrompt4LJP significantly decreased by 0.05 in the shot-8 scenario, which indicates that the knowledgeable prompt plays an essential role in the few-shot scenario. Next, we removed the soft prompt tokens from the prompt template. As seen from the table, this caused a slight decrease in KnowPrompt4LJP's macro F1 value in the shot-8 scenario but did not change the macro F1 value in the full-size training data scenario. This indicates that soft prompt tokens have a role but a limited role. This may be because a small number of soft prompt tokens cannot play a dominant role relative to a case description text with a large number of tokens.

Finally, we replaced the checkpoint Lawformer used in KnowPrompt4LJP with the regular BERT checkpoint. We found that this had a dramatic impact on KnowPrompt4LJP. The macro F1 value of KnowPrompt4LJP in the shot-8 condition was significantly lower by 0.06, while its macro F1 value on the full-size training data was significantly lower by 0.09. This result is not unexpected since replacing Lawformer with the regular BERT checkpoint forced us to reduce the maximum input length of KnowPrompt4LJP from 2,000 to 512. In the CAIL2018 dataset, there are more than 200,000 case description texts longer than 512, so most of the text could not be completely characterized. We can see that the model's results on the full-size training data scenario after using regular BERT checkpoints are even lower than the results of CNN, Attention, and HBERT. This indicates that a complete semantic representation of a case description text is decisive for the final results.

Moreover, even if the maximum input length of Lawformer is also set to 512, using Lawformer is still 0.04 higher than using regular BERT checkpoints in terms of macro F1 values (in the full-size training data scenario). This fully demonstrates that Lawformer, pre-trained on a Chinese legal corpus, is closer to the semantic distribution of the case description texts in the LJP task than the BERT checkpoint pre-trained on the Wikipedia corpus.

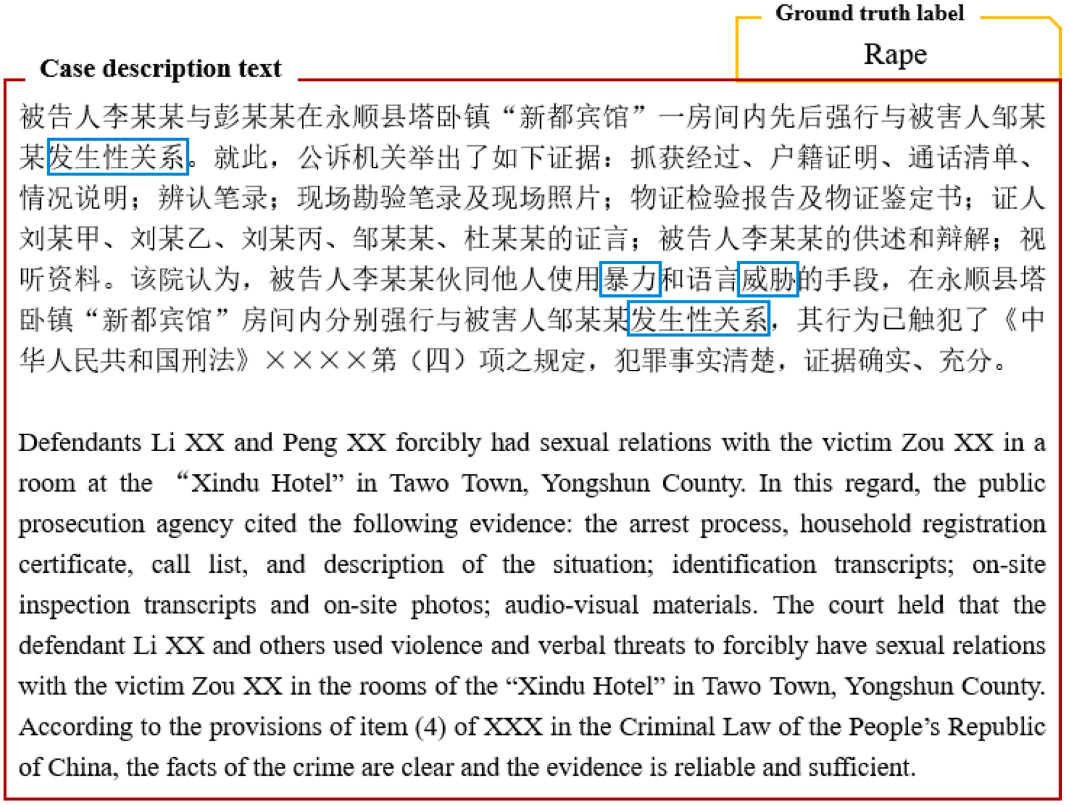
6.3. *Interpretability analysis*

In this section, we qualitatively analyze the advantages of KnowPrompt4LJP in terms of interpretability through a case. Interpretability is crucial in the legal judgement prediction task. People will not fully trust the answers given by computers unless they give reasonable explanations. The case is illustrated in Fig. 6. The central part of the figure shows a case description text, and the upper right corner shows the ground truth label corresponding to the text. As can be seen from the figure, three keywords are extracted from the case description text via the external knowledge base: 发生性关系(Sexual relation), 暴力(Violence), and 恐吓(Threat). Intuitively, we can agree that these three keywords are highly related to the crime of rape. Therefore, it can be said that the keywords extracted via the external knowledge base provide evidence for KnowPrompt4LJP's prediction. This demonstrates that our proposed KnowPrompt4LJP has a certain degree of interpretability.

Next, we observe the predictions of KnowPrompt4LJP for the case. Fig. 7 shows the top three tokens at the first four mask positions in terms of scoring. Mask 1 in the figure indicates the first masked position. It can be seen that the highest-scored token in the first masked position is "强", and the second and third-scored tokens are "胁" and "诱", respectively. Mask 2 in the figure indicates the second masked position. As we can see, the highest-scored token in the second masked position is "奸", and the second and third-scored tokens are "淫" and "性" respectively. The combination of the highest-scored tokens in Mask 1 and Mask 2 forms the word "强奸(Rape)", which is consistent with the ground truth label of the case description. In addition, we observed that the tokens "胁",

Table 6
Ablation experiment results.

	Shot-8			Full scale		
	MacroP	MacroR	MacroF1	MacroP	MacroR	MacroF1
KnowPrompt4LJP	0.72	0.69	0.70	0.82	0.81	0.81
Remove ***knowledgeable prompt***	0.66	0.65	0.65 (−0.05)	0.81	0.78	0.79 (−0.02)
Remove ***soft prompt token***	0.65	0.62	0.63 (−0.02)	0.80	0.78	0.79 (−0.00)
Replace ***Lawformer*** with ***regular BERT checkpoint***	0.58	0.57	0.57 (−0.06)	0.73	0.68	0.70 (−0.09)



Ground truth label
Rape
Case description text

被告人李某某与彭某某在永顺县塔卧镇“新都宾馆”一房间内先后强行与被害人邹某某发生性关系。就此，公诉机关举出了如下证据：抓获经过、户籍证明、通话清单、情况说明；辨认笔录；现场勘验笔录及现场照片；物证检验报告及物证鉴定书；证人刘某甲、刘某乙、刘某丙、邹某某、杜某某的证言；被告人李某某的供述和辩解；视听资料。该院认为，被告人李某某伙同他人使用暴力和语言威胁的手段，在永顺县塔卧镇“新都宾馆”房间内分别强行与被害人邹某某发生性关系，其行为已触犯了《中华人民共和国刑法》××××第（四）项之规定，犯罪事实清楚，证据确实、充分。

Defendants Li XX and Peng XX forcibly had sexual relations with the victim Zou XX in a room at the “Xindu Hotel” in Tawo Town, Yongshun County. In this regard, the public prosecution agency cited the following evidence: the arrest process, household registration certificate, call list, and description of the situation; identification transcripts; on-site inspection transcripts and on-site photos; audio-visual materials. The court held that the defendant Li XX and others used violence and verbal threats to forcibly have sexual relations with the victim Zou XX in the rooms of the “Xindu Hotel” in Tawo Town, Yongshun County. According to the provisions of item (4) of XXX in the Criminal Law of the People’s Republic of China, the facts of the crime are clear and the evidence is reliable and sufficient.

Fig. 6. Case for interpretability analysis.

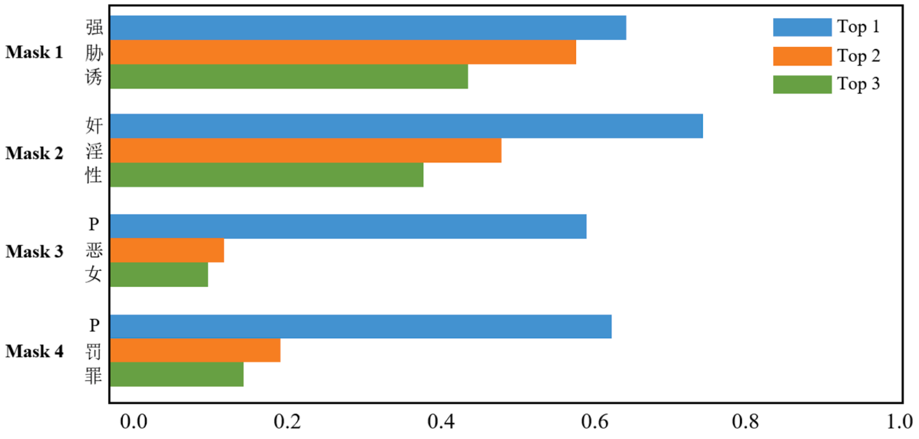


Fig. 7. Top-3 scored predicted tokens at the first four mask positions.

“诱”, “淫”, and “性”, which scored high in Mask 1 and Mask 2 positions, were all associated with the occurrence of rape.

Begin from the third masked token Mask 3, the highest score predicted by the model is “P”, which represents token [PAD]. Therefore, the final output of KnowPrompt4LJP was “强奸[P][P][P][P][P][P][P][P]”. The final result was “强奸(Rape)” after removing the padding tokens. The Jaccard similarity between “强奸” and the 196 accusations was calculated, and the correct result was obtained. Throughout the process, it can be seen that KnowPrompt4LJP can obtain accurate results on the mask language model task, which ensured the accuracy of the subsequent label mapping. In addition, we also see that KnowPrompt4LJP can accurately predict padding tokens when the accusation label text is short.

6.4. Generalization analysis

Although KnowPrompt4LJP is proposed on a Chinese LJP dataset, the model can also be applied to datasets in other languages as long as different tokenizers and backbones are used. To demonstrate this, this section tests the performance of KnowPrompt4LJP on other datasets.

The ECHR dataset was constructed by (Chalkidis et al., 2019) based on the European Convention of Human Rights and contains more than 10,000 cases. Each case is mapped to articles of the Convention that were violated (if any). As the author states, in the ECHR dataset, 45 out of 66 labels are not present in the training set, while another 11 are present in fewer than 50 cases. Therefore, the ECHR dataset can test the abilities of models in few-shot scenarios.

Table 7 shows the performance of the baselines and KnowPrompt4LJP on the ECHR dataset. It is worth noting that we have not found a suitable external knowledge base for this dataset, so we delete the knowledge injection module of KnowPrompt4LJP in this experiment. In other words, we use a castrated version of KnowPrompt4LJP for unfair comparisons with the baselines. It can be seen from the table that KnowPrompt4LJP can still achieve the best performance even if its knowledge injection module is dropped. This indicates that KnowPrompt4LJP also has significant advantages of few-shot in English LJP datasets. More specifically, it is prompt learning that improves the performance of the LJP task. Therefore, we can say that prompt learning is beneficial for legal AI tasks (Tables 8 and 9).

In addition, we test the performance of KnowPrompt4LJP on the ILSI dataset proposed by (Paul et al., 2022) and the ISCJD dataset proposed by (Paul et al., 2020). The ILSI dataset is constructed based on criminal case documents and statutes from the Indian judiciary. It contains 42,884 training documents, 10,203 validation documents, and 13,043 test documents. ISCJD is constructed based on Indian Penal Code, which consider the 20 most frequent charges. As can be seen from the table, KnowPrompt4LJP still shows strong advantages compared with other baselines on these two datasets. It is worth noting that we also removed the knowledge injection module of KnowPrompt4LJP because we have not yet found a suitable knowledge base. The experimental results demonstrate the good generalization of KnowPrompt4LJP. In addition, we believe that the performance of KnowPrompt4LJP on the English datasets will be further improved in the future if external knowledge bases suitable for the datasets are found.

7. Discussion

In this paper, we propose a prompt learning framework-based LJP method called KnowPrompt4LJP, which not only improves SOTA results on the Chinese LJP dataset under the condition of using full-size training data but also achieves superior performance well beyond the baselines in zero-shot and few-shot scenarios. KnowPrompt4LJP is inspired by the recently popular prompt learning methodology and adapted to the characteristics of the LJP task. Moreover, we believe KnowPrompt4LJP is helpful for future works on the LJP task. In this section, we first describe the connections and differences between KnowPrompt4LJP and existing works. Subsequently, we will explore the contribution of KnowPrompt4LJP to future work on the LJP tasks. Finally, we will analyze the impact of KnowPrompt4LJP on the system design.

Table 7
Performance of the methods on the ECHR dataset.

Method	MacroP	MacroR	MacroF1
CNN	0.60	0.50	0.55
BiLSTM	0.62	0.51	0.56
BiGRU	0.59	0.49	0.54
Attention	0.63	0.54	0.58
BERT_FT	0.40	0.27	0.32
HBERT	0.66	0.55	0.60
KnowPrompt4LJP (ours) without knowledge injection	0.68	0.57	0.62

Table 8
Performance of the methods on the ILSI dataset.

Method	MacroP	MacroR	MacroF1
CNN	0.04	0.36	0.07
BiLSTM	0.05	0.49	0.09
BiGRU	0.05	0.42	0.09
Attention	0.10	0.54	0.17
BERT_FT	0.02	0.34	0.04
HBERT	0.04	0.53	0.07
KnowPrompt4LJP (ours) without knowledge injection	0.04	0.65	0.08

Table 9
Performance of the methods on the ISCJD dataset.

Method	MacroP	MacroR	MacroF1
CNN	0.18	0.69	0.29
BiLSTM	0.23	0.74	0.35
BiGRU	0.19	0.73	0.30
Attention	0.22	0.78	0.34
BERT_FT	0.24	0.48	0.32
HBERT	0.60	0.54	0.57
KnowPrompt4LJP (ours) without knowledge injection	0.62	0.57	0.59

7.1. Connection and comparison with existing works

The existing LJP methods can be classified into rule-based, shallow machine learning-based, and deep learning-based methods. KnowPrompt4LJP is essentially a deep learning-based method because its inference kernel is a pre-trained deep language model. Compared with rule-based methods and shallow machine learning-based methods, deep learning-based methods tend to have stronger generalizations and semantic representations. However, these methods can only achieve good results with sufficient labeled data. In contrast, thanks to the prompt learning framework, KnowPrompt4LJP can achieve satisfactory results in different data scenarios.

7.2. Contributions to future research

LJP is a highly domain-specific task. Compared with other general-purpose natural language processing tasks, there are not enough large-scale annotated datasets for the LJP task. To the best of our knowledge, there are only two large-scale available datasets for the LJP task: CAIL2018 (Chinese) and ECHR (English). Also, CAIL2018 only contains descriptions of criminal cases, and ECHR only contains human rights cases. Some other fine-grained fields, such as taxation, civil disputes, economic disputes, administrative litigation, labor arbitration, etc. do not have large-scale labeled data yet. In fact, it is difficult to build large-scale labeled data for each fine-grained sub-field due to the expensive labeling cost.

KnowPrompt4LJP can achieve the same results as using large-scale labeled data with only a tiny amount of labeled training data. This allows researchers to implement LJP tasks on other fine-grained fields in the future using only a small amount of labeled data. This facilitates the application of LJP tasks to other sub-fields. Furthermore, the prompt learning architecture in KnowPrompt4LJP and the idea of knowledge incorporation can be easily generalized to other tasks in legal AI, such as legal provisions recommendation, court opinion prediction, legal question classification, etc.

7.3. Implications on system design

A realistic LJP system must cope with case descriptions on various sub-fields. However, annotated training data are often scarce in some rare sub-fields, such as administrative litigation, while KnowPrompt4LJP provides a feasible solution for realistic LJP systems to be comfortable with different data resource scenarios. In addition, a reliable LJP system should stock interpretable knowledge. The knowledgeable prompt in KnowPrompt4LJP can effectively identify keywords in case descriptions by mounting an external knowledge base, thus giving an LJP system the potential to provide interpretable results to users.

8. Conclusion and future work

In this paper, we propose a novel LJP method, called KnowPrompt4LJP, based on prompt learning frameworks. The method can stimulate the PLM's recall of the knowledge learned in pre-training and enhances the PLM's understanding of LJP tasks through a well-designed prompt template. We use Lawformer as the PLM checkpoint used in the method so that the maximum input to the model is larger than 512. This allows the model to completely characterize a long case description text's semantics. In addition, we incorporated keyword information from case description texts into the prompt template through an external knowledge base to further enhance the guidance of the prompt template to the PLM.

To validate the effectiveness of KnowPrompt4LJP, we conducted extensive experiments on CAIL2018, a high-quality Chinese LJP dataset. The experimental results show that KnowPrompt4LJP can achieve results beyond the baselines in both low-resource and data-rich scenarios. In particular, our method can still achieve a macro F1 value of 0.7 in the low-resource scenario, comparable to the results of the baselines in the data-rich scenario. In addition, the experiments demonstrate the contribution of the external knowledge base to the method, which can significantly improve PLM's understanding of case descriptions.

Tasks that are not yet can achieve satisfactory results in low-resource scenarios still exist in the field of Legal AI. In the future, we will extend the idea of prompt learning to other legal AI tasks, such as legal provisions recommendation, similar case retrieval, legal Q&A, etc. Moreover, interpretability is more critical for tasks in the legal domain than for tasks in the general-purpose domain. So, we will explore how to improve the interpretability of the results under the prompt learning framework next.

CRediT authorship contribution statement

Jingyun Sun: Conceptualization, Methodology, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Resources. **Shaobin Huang:** Conceptualization, Supervision. **Chi Wei:** Formal analysis, Investigation, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

Thanks to anonymous reviewers for their constructive comments on this paper. Their suggestions have played an important role in improving the quality of this paper.

Funding

The authors did not receive support from any organization for the submitted work.

Availability of data and material

The data in this paper are available by contacting the corresponding authors.

Code availability

The code in this paper is available by contacting the corresponding authors.

References

Barros, R., Peres, A., Lorenzi, F., Krug Wives, L., & Hubert da Silva Jaccottet, E. (2018). Case law analysis with machine learning in Brazilian court. *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer.

Bhattacharya, P., Ghosh, K., Pal, A., & Ghosh, S. (2022). Legal case document similarity: You need both network and text. *Information Processing & Management, 59*(6), Article 103069.

Chalkidis, I., I. Androutsopoulos and N. Aletras (2019). Neural Legal Judgment Prediction in English, Florence, Italy, Association for Computational Linguistics.

Chen, H., D. Cai, W. Dai, Z. Dai and Y. Ding (2019). Charge-Based Prison Term Prediction with Deep Gating Network. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).

Chen, X., X. Xie, N. Zhang, J. Yan, S. Deng, C. Tan, F. Huang, L. Si and H. Chen (2021). "AdaPrompt: Adaptive Prompt-based Finetuning for Relation Extraction. CoRR abs/2104.07650 (2021)." arXiv preprint arXiv:2104.07650.

Chung, J., C. Gulcehre, K. Cho and Y. Bengio (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. NIPS 2014 Workshop on Deep Learning, December 2014.

Clark, P., O. Tafjord and K. Richardson (2021). Transformers as soft reasoners over language. Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence. Yokohama, Yokohama, Japan: Article 537.

Ding, N., Y. Chen, X. Han, G. Xu, P. Xie, H.-T. Zheng, Z. Liu, J. Li and H.-G. Kim (2021). "Prompt-learning for fine-grained entity typing." arXiv preprint arXiv:2108.10604.

Ding, N., S. Hu, W. Zhao, Y. Chen, Z. Liu, H. Zheng and M. Sun (2022). OpenPrompt: An Open-source Framework for Prompt-learning, Dublin, Ireland, Association for Computational Linguistics.

Fang, J., Li, X., & Liu, Y. (2022). Low-Resource Similar Case Matching in Legal Domain. *International Conference on Artificial Neural Networks*. Springer.

Fawei, B., J. Z. Pan, M. Kollingbaum and A. Z. Wyner (2018). A methodology for a criminal law and procedure ontology for legal question answering. Joint International Semantic Technology Conference, Springer.

Feng, Y., C. Li and V. Ng (2022). Legal Judgment Prediction: A Survey of the State of the Art. Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22.

Gao, T., A. Fisch and D. Chen (2021). Making Pre-trained Language Models Better Few-shot Learners, Online, Association for Computational Linguistics.

Han, X., W. Zhao, N. Ding, Z. Liu and M. Sun (2021). "Ptr: Prompt tuning with rules for text classification." arXiv preprint arXiv:2105.11259.

Hendrycks, D., Burns, C., Chen, A., & Ball, S. (2021). CUAD: An Expert-Annotated NLP Dataset for Legal Contract Review. *NeurIPS*.

Hong, J., C. Voss and C. D. Manning (2021). Challenges for information extraction from dialogue in criminal law. Proceedings of the 1st Workshop on NLP for Positive Impact.

Hu, S., N. Ding, H. Wang, Z. Liu, J. Wang, J. Li, W. Wu and M. Sun (2022). Knowledgeable Prompt-tuning: Incorporating Knowledge into Prompt Verbalizer for Text Classification. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).

Hu, Z., X. Li, C. Tu, Z. Liu and M. Sun (2018). Few-shot charge prediction with discriminative legal attributes. Proceedings of the 27th International Conference on Computational Linguistics.

Lauderdale, B. E., & Clark, T. S. (2012). The Supreme Court's many median justices. *American Political Science Review, 106*(4), 847–866.

Lester, B., R. Al-Rfou and N. Constant (2021). The Power of Scale for Parameter-Efficient Prompt Tuning. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing.

Li, J., G. Zhang, H. Yan, L. Yu and T. Meng (2018). A Markov logic networks based method to predict judicial decisions of divorce cases. 2018 IEEE International Conference on Smart Cloud (SmartCloud), IEEE.

Li, R., Li, Z., Huang, S., Liu, Y., & Qiu, J. (2021). TransExplain: Using neural networks to find suitable explanations for Chinese phrases. *Expert Systems with Applications, 115440*.

Li, X. L. and P. Liang (2021). Prefix-Tuning: Optimizing Continuous Prompts for Generation, Online, Association for Computational Linguistics.

Liu, P., W. Yuan, J. Fu, Z. Jiang, H. Hayashi and G. Neubig (2021). “Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing.” arXiv preprint arXiv:2107.13586.

LiuC, C., & HoJ. (2004). Case instance generation and refinement for case-based criminal summary judgments in Chinese. *Journal of Information Science and Engineering, 20*(4), 283–800.

Loukas, L., Fergadiotis, M., Chalkidis, I., Spyropoulou, E., Malakasiotis, P., Androutsopoulos, I., & Paliouras, G. (2022). *FiNER: Financial Numeric Entity Recognition for XBRL Tagging*. ACL.

Lyu, Y., Wang, Z., Ren, Z., Ren, P., Chen, Z., Liu, X., … Song, H. (2022). Improving legal judgment prediction through reinforced criminal element extraction. *Information Processing & Management, 59*(1), Article 102780.

Ma, L., Y. Zhang, T. Wang, X. Liu, W. Ye, C. Sun and S. Zhang (2021). Legal Judgment Prediction with Multi-Stage Case Representation Learning in the Real Court Setting. Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval.

Mandal, A., K. Ghosh, S. Ghosh and S. Mandal (2021). “A sequence labeling model for catchphrase identification from legal case documents.” Artificial Intelligence and Law: 1-34.

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

Paul, S., Goyal, P., & Ghosh, S. (2020). Automatic charge identification from facts: A few sentence-level charge annotations is all you need. *Proceedings of the 28th International Conference on Computational Linguistics*.

Paul, S., Goyal, P., & Ghosh, S. (2022). LeSICiN: A heterogeneous graph-based approach for automatic legal statute identification from Indian legal documents. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Qin, G., & Eisner, J. (2021). *Learning How to Ask: Querying LMs with Mixtures of Soft Prompts*. Association for Computational Linguistics: Online.

Roberts, A., Raffel, C., & Shazeer, N. (2020). How Much Knowledge Can You Pack Into the Parameters of a Language Model?. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Schick, T. and H. Schütze (2021). Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference, Online, Association for Computational Linguistics.

Segal, J. A. (1984). Predicting Supreme Court cases probabilistically: The search and seizure cases, 1962–1981. *American Political Science Review, 78*(4), 891–900.

Shaghaghian, S., L. Y. Feng, B. Jafarpour and N. Pogrebnyakov (2020). Customizing contextualized language models for legal document reviews. 2020 IEEE International Conference on Big Data (Big Data), IEEE.

Shin, R., C. Lin, S. Thomson, C. Chen, S. Roy, E. A. Platanios, A. Pauls, D. Klein, J. Eisner and B. Van Durme (2021). Constrained Language Models Yield Few-Shot Semantic Parsers, Online and Punta Cana, Dominican Republic, Association for Computational Linguistics.

Sukanya, G., & Priyadarshini, J. (2021). A Meta Analysis of Attention Models on Legal Judgment Prediction System. *International Journal of Advanced Computer Science and Applications, 12*(2).

Talmor, A., Elazar, Y., Goldberg, Y., & Berant, J. (2020). oLMpics-On What Language Model Pre-training Captures. *Transactions of the Association for Computational Linguistics, 8*, 743–758.

Wang, H., T. He, Z. Zou, S. Shen and Y. Li (2019). Using case facts to predict accusation based on deep learning. 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C), IEEE.

Wang, P., Fan, Y., Niu, S., Yang, Z., Zhang, Y., & Guo, J. (2019). Hierarchical matching network for crime classification. *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*.

Xiao, C., Hu, X., Liu, Z., Tu, C., & Sun, M. (2021). Lawformer: A pre-trained language model for chinese legal long documents. *AI Open, 2*, 79–84.

Xiao, C., H. Zhong, Z. Guo, C. Tu, Z. Liu, M. Sun, Y. Feng, X. Han, Z. Hu and H. Wang (2018). “Cail2018: A large-scale legal dataset for judgment prediction.” arXiv preprint arXiv:1807.02478.

Yang, Z., P. Wang, L. Zhang, L. Shou and W. Xu (2019). A recurrent attention network for judgment prediction. International Conference on Artificial Neural Networks, Springer.

, L., J. Wang, S. Fan, Y. Bian, B. Yang, Y. Wang and X. Wang (2019). Automatic Legal Judgment Prediction via Large Amounts of Criminal Cases. 2019 IEEE 5th International Conference on Computer and Communications (ICCC), IEEE.

Yue, L., Liu, Q., Jin, B., Wu, H., Zhang, K., An, Y., … Wu, D. (2021). NeurJudge: A circumstance-aware neural framework for legal judgment prediction. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Zhong, H., Guo, Z., Tu, C., Xiao, C., Liu, Z., & Sun, M. (2018). Legal judgment prediction via topological learning. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Zhong, H., Wang, Y., Tu, C., Zhang, T., Liu, Z., & Sun, M. (2020). Iteratively questioning and answering for interpretable legal judgment prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Zhong, H., C. Xiao, C. Tu, T. Zhang, Z. Liu and M. Sun (2020). How Does NLP Benefit Legal System: A Summary of Legal Artificial Intelligence, Online, Association for Computational Linguistics.

Zhong, H., Xiao, C., Tu, C., Zhang, T., Liu, Z., & Sun, M. (2020b). JEC-QA: A legal-domain question answering dataset. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Zhu, T., Y. Qin, Q. Chen, B. Hu and Y. Xiang (2022). “Enhancing Entity Representations with Prompt Learning for Biomedical Entity Linking.