Applied Soft Computing Journal 180 (2025) 113438

Contents lists available at ScienceDirect

Applied Soft Computing

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

A multi-source heterogeneous knowledge injected prompt learning method for legal charge prediction

Jingyun Sun [a,*], Chi Wei [b]

[a] College of Computer and Control Engineering, Northeast Forestry University, Harbin, China
[b] Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China

HIGHLIGHTS

- Enhancing legal charge prediction using multi-source heterogeneous legal knowledge.
- Encapsulating multi-source heterogeneous legal knowledge through a prompt-learning framework.
- Extracting factual elements via a conversational LLM.
- Employing a legal knowledge base to match legal knowledge snippets.

ARTICLE INFO

Keywords:
Legal charge prediction
Knowledge-enhanced prompt learning
Contrastive learning

ABSTRACT

Legal charge prediction, an essential task in legal AI, seeks to assign accurate charge labels to case descriptions, attracting significant recent interest. Existing methods primarily employ diverse neural network structures for modeling case descriptions directly, failing to effectively leverage multi-source external knowledge. Therefore, we propose a prompt learning framework-based method that simultaneously leverages multi-source heterogeneous external legal knowledge. Specifically, we match knowledge snippets in case descriptions via the legal knowledge base and encapsulate them into the input through a hard prompt template. Additionally, we retrieve legal articles related to the given case description through contrastive learning, and then obtain factual elements through a conversational Large Language Model (LLM). We fuse the embedding vectors of soft prompt tokens with the encoding vector of factual elements to achieve knowledge-enhanced model forward inference. Experimental results show that our method achieved state-of-the-art results on CAIL-2018, the largest legal charge prediction dataset, and our method has lower data dependency. Case studies also demonstrate our method's strong interpretability.

1. Introduction

Legal charge prediction is a crucial task in legal artificial intelligence, aimed at utilizing advanced technologies such as machine learning, deep learning, and natural language processing to analyze given case descriptions and thereby predict corresponding charge labels. Fig. 1 presents an example of the task. The figure provides a case description, from which it can be inferred that the case pertains to a mobile phone theft. Consequently, the "theft" label is selected from the candidate legal charge labels to be assigned to this case description.

Legal charge prediction not only helps legal professionals handle cases efficiently and accurately, reducing human errors and increasing the consistency and fairness of judgments, but also supports the enhancement of legal education and public legal awareness, by spreading legal knowledge and strengthening society's understanding and compliance with legal regulations.

Legal charge prediction is often regarded as a classification problem, hence researchers typically adopt methods similar to those used for general text classification tasks to address it. For instance, Wang et al. proposed a convolutional neural networks-based approach [1], Yang et al. introduced a method based on bidirectional long short-term memory network [2], and Chen et al. developed a gated recurrent units-based method [3].

However, legal charge prediction differs from general text classification tasks in several ways. Firstly, legal texts, which contain a plethora of legal terminologies and keywords, are distinct from general texts,

* Corresponding author.
E-mail address: sunjingyun@nefu.edu.cn (J. Sun).

https://doi.org/10.1016/j.asoc.2025.113438
Received 17 March 2024; Received in revised form 24 May 2025; Accepted 5 June 2025
Available online 6 June 2025

presenting challenges to universal models in understanding the content. Secondly, legal charge prediction focuses more on the factual information within texts, whereas general text classification tasks are concerned with the topics described by texts. Therefore, many researchers have utilized language models pretrained in the legal domain as the backbone to enhance the model's comprehension of legal texts [4]. Such models are better at capturing the domain-specific terminologies and keywords within legal texts. Nevertheless, the pretrained models they employed only allow for the input of 512 tokens, which is insufficient for modeling legal texts that often exceed this length.

Moreover, to capture factual information in legal texts, Sukanya & Priyadarshini et al. proposed a model based on attention mechanism [5]. Some studies introduced hierarchical attention mechanisms to capture factual information within case descriptions in a layered manner, addressing the limitations imposed by input length [6,7]. However, they only utilized the content of legal texts themselves to obtain factual information, without leveraging external structured knowledge.

In contrast, the method we propose not only enhances the model's comprehension of the textual content and task objectives but also fully leverages heterogeneous external legal knowledge from multiple sources. As Tang et al. [8] have noted, domain knowledge can significantly enhance a model's understanding of the current task.

Firstly, we employ the newly introduced pre-trained language model, Lawformer, which is trained on a large-scale legal corpus and can accommodate text inputs exceeding 4000 tokens, as our inference model. Lawformer aids in accurately comprehending the semantics of case descriptions and capturing the meanings expressed by legal terms and keywords. Subsequently, we utilize a legal knowledge base to match knowledge snippets from case descriptions, while employing a conversational Large Language Model (LLM) and relevant articles to extract factual elements from descriptions. This process introduces external components to assist the model in acquiring legal knowledge, thereby further enhancing the model's understanding of case descriptions. Finally, we propose using soft prompt tokens and hard prompt templates to encapsulate heterogeneous legal knowledge from multiple sources. Overall, the method presented in this paper leverages the paradigm of prompt learning to integrate heterogeneous legal knowledge from multiple sources into the model's forward reasoning, thus improving the model's predictive accuracy regarding legal charges.

We conducted extensive experiments on CAIL-2018, the largest legal charge prediction dataset to date [9]. The results demonstrate that our proposed method achieved results surpassing the baselines, with a macro F1 score of 0.84. Moreover, the experiments indicate that our method has the lowest dependency on training data. The performance of other baselines significantly diminishes as the scale of training data decreases, whereas our method still maintains a high F1 score. We also analyzed the contribution of each module within our method through ablation studies. Finally, we validated that our approach possesses strong interpretability, which is crucial for artificial intelligence tasks in the legal domain.

Our primary contributions can be summarized as follows:

1) We propose a legal charge prediction model that integrates multi-source heterogeneous legal knowledge.
2) We introduce a method to encapsulate heterogeneous legal knowledge via the prompt-based learning framework.
3) We propose the use of a conversational LLM and relevant legal articles to extract factual elements from case descriptions.
4) We employ a specialized legal knowledge base to match knowledge snippets from case descriptions.
5) We demonstrate the effectiveness of our method through extensive empirical validation.

The remainder of this paper is organized as follows: Section 2 reviews related work and identifies the research gap; Section 3 provides a formal definition of the task; Section 4 presents our proposed method; Section 5 details the experimental setup; Section 6 showcases the experimental results and provides an analysis; Section 7 concludes the paper.

2. Related work

This section presents the works related to our study. Firstly, the latest progress in legal charge prediction is introduced. Subsequently, the basic concepts and applications of prompt learning are introduced.

2.1. Legal charge prediction

Legal charge prediction is a crucial task in the field of legal artificial intelligence, aimed at predicting the legal charges corresponding to given case descriptions[10,11]. Due to the scarcity of legal resources, individuals lacking legal knowledge often find it challenging to promptly seek legal advice from attorneys or legal professionals. Therefore, the automation of legal charge prediction can, to a certain extent, alleviate the issue of legal resource scarcity. Furthermore, legal charge prediction can also offer decision support for lawyers or judges, thereby enhancing their work efficiency.

Early legal charge predictions primarily relied on rule-based methods or mathematical models [12]. These methods have the advantage of transparent and intuitive reasoning processes, and once the inference rules are triggered, their outcomes are fixed. However, such methods exhibit poor generalization and struggle to effectively address language phenomena such as synonyms and polysemy in case descriptions. With the introduction of the Word2Vec concept by Mikolov

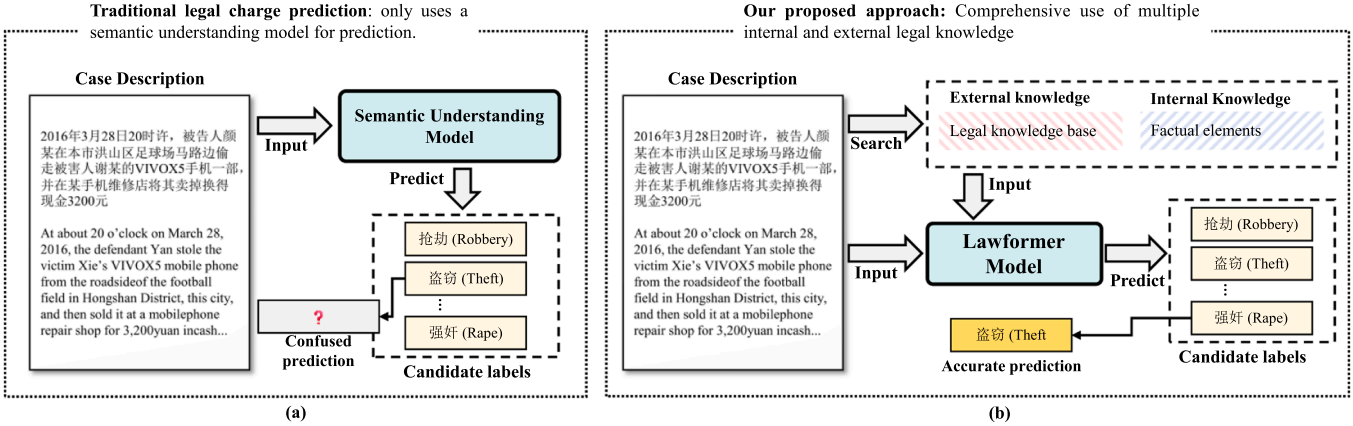

Fig. 1. An example of the legal charge prediction task.

[13], subsequent legal charge prediction methods have predominantly been based on semantic embeddings. These methods embed words in case descriptions into semantic vectors, which are then used as features inputted into a machine learning model [4,14]. An exemplary contribution is Law2Vec proposed by Chalkidis et al. [15]. Additionally, methods combining semantic embeddings with support vector machines and k-nearest neighbors have been proposed [16].

With the popularity of deep learning technologies, researchers have shifted from merely combining semantic embeddings with shallow machine learning models to integrating them with neural network models [4,17–19]. Compared to shallow machine learning models, neural network models exhibit stronger data fitting and feature learning capabilities, leading to superior performance. For instance, Wang et al. proposed a method based on convolutional neural networks [1]. This method leverages the modeling capability of convolutional networks for local key information, enabling the model to identify crucial phrases, terms, jargon, and keywords in case descriptions, thereby enhancing the performance for legal charge prediction. On the other hand, Yang et al. focused on modeling the global semantic correlations within case descriptions and introduced a method based on long short-term memory networks [2]. To reduce the computational complexity of globally modeling semantic relationships in case descriptions, Chen et al. presented a method based on gated recurrent units [3]. This method employs computationally less intensive gated recurrent units to model the global semantic correlations of case descriptions, significantly reducing computational complexity. Additionally, Sukanya & Priyadarshini proposed a model based on attention, which can attend to salient information in different aspects of a case description [5]. Building upon this, Wang et al. introduced a hierarchical attention model capable of attending to salient information at different levels within a case description [20]. This approach achieved state-of-the-art performance in various benchmarks. Despite the generally excellent results achieved by neural network-based models, they are still constrained by the effectiveness of word embeddings and rely heavily on large amounts of high-quality annotated data.

In recent years, an increasing number of methods for predicting legal charges have been based on pre-trained language models [4,21]. Unlike neural network models that take static semantic embeddings as input, pre-trained language models are pre-trained on large-scale textual data and therefore have better context understanding capabilities [22–24]. Moreover, pre-trained models can capture relationships and contexts between different words in a case description, thereby better comprehending the textual meaning of legal narratives and contributing to improved accuracy in charge prediction [6,25]. The core of our method lies in a pre-trained legal language model, ensuring its contextual understanding of legal texts. Diverging from existing methods, we integrate heterogeneous legal knowledge from multiple sources into the reasoning process of the language model, enabling it to acquire a more comprehensive understanding of legal knowledge specific to a given case description. Furthermore, compared to traditional neural network methods, our approach exhibits lower data dependency.

2.2. Prompt learning

Prompt learning has recently garnered significant attention from researchers due to its ability to stimulate language models to better recall the semantic knowledge learned during pre-training [26,27]. Unlike the standard downstream task fine-tuning paradigm, the prompt learning paradigm aligns downstream tasks with the pre-training tasks of language models. To this end, methods based on prompt learning should first convert different downstream tasks into a language modeling task [28,29]. For instance, a traditional classification task is designed to fit the probability distribution $Y = (X;\theta)$. Given a piece of text $x = [\textit{This pizza is so delicious}]$, the model might output the prediction $y = 0 \in \{0,1,2\}$ once θ is learned. Where 0 denotes a positive sentiment label, 1 denotes a negative sentiment label, and 2 denotes a neutral sentiment label. However, the model aims to fit the function $Y = P(MASK = \mathscr{V}_y|template(X);\theta)$ by converting the task into a language modeling task. Here, $template(x)$ is a new text transformed from the original text by inserting specific prompt words and $\mathscr{V}_y$ is the set of label words. For example, the original text $x = [\textit{This pizza is so delicious}]$ could be transformed into $template(x)$= [This pizza is so delicious. It feels [MASK]], and the model is tasked with predicting the word at the [MASK] position based on θ, thereby inferring the sentiment label. The model might generate words such as "amazing", "bad", or "okay", which can then be mapped to the specific sentiment labels 0, 1, or 2.

The three core components in prompt learning are prompt templates, inference models, and label mappings. A ***prompt template*** is employed to encapsulate a original text into a new format featuring prompts and masks, as exemplified in the work of [30–32]. They integrate external knowledge at the prompt template stage to maximize the language model's understanding of the task. Our method also incorporates external knowledge during the prompt template stage. However, we propose the integration of multi-source heterogeneous knowledge into the prompt template unlike existing methods, thereby significantly enhancing the model's inference capabilities. ***Inference model*** is a core component of prompt learning, utilized for predicting the tokens at the mask positions based on the encapsulated text. Commonly used inference models include BERT [33], RoBERTa [34], and the T5 series [35]. Furthermore, some studies employ language models specialized in specific domains to cater to particular tasks. For instance, Zhu et al. proposed using CliniBERT [36] for prompt learning methods in the medical field, achieving state-of-the-art results. We utilize Lawformer, a pre-trained language model in the legal domain, as the inference model, making our method more suitable for legal charge prediction tasks.

2.3. Topic modeling

Topic modeling, an unsupervised learning technique, can automatically uncover latent thematic structures from textual data and illuminate semantic patterns in document collections. It has been widely adopted in domains such as news [37], social media [38], and academic literature [39], and also plays a pivotal role in legal text analysis by facilitating case comprehension, aiding in charge prediction, and identifying legal trends.

Early topic modeling methods primarily relied on word embedding techniques, which map words into low-dimensional vector spaces to capture semantic relationships. This approach groups words that are close in the embedding space into the same topic. For instance, "theft" and "robbery" are semantically similar and thus can be classified under the same topic. Some legal text classification methods first use word embeddings to obtain semantic representations, then integrate clustering, hidden topic analysis, or other topic modeling approaches for classification. For example, Wang et al. [40] proposed a legal text classification method based on Latent Dirichlet Allocation (LDA), while Rawat et al. [41] combined LDA with Latent Semantic Analysis (LSA) to further improve classification performance.

With the rapid development of deep learning technologies, deep learning-based topic modeling methods have gradually become the mainstream approach for legal text classification. Compared to traditional word embedding methods, these approaches leverage neural network architectures to extract deeper hidden features from the text, further enhancing the understanding of text semantics. For instance,

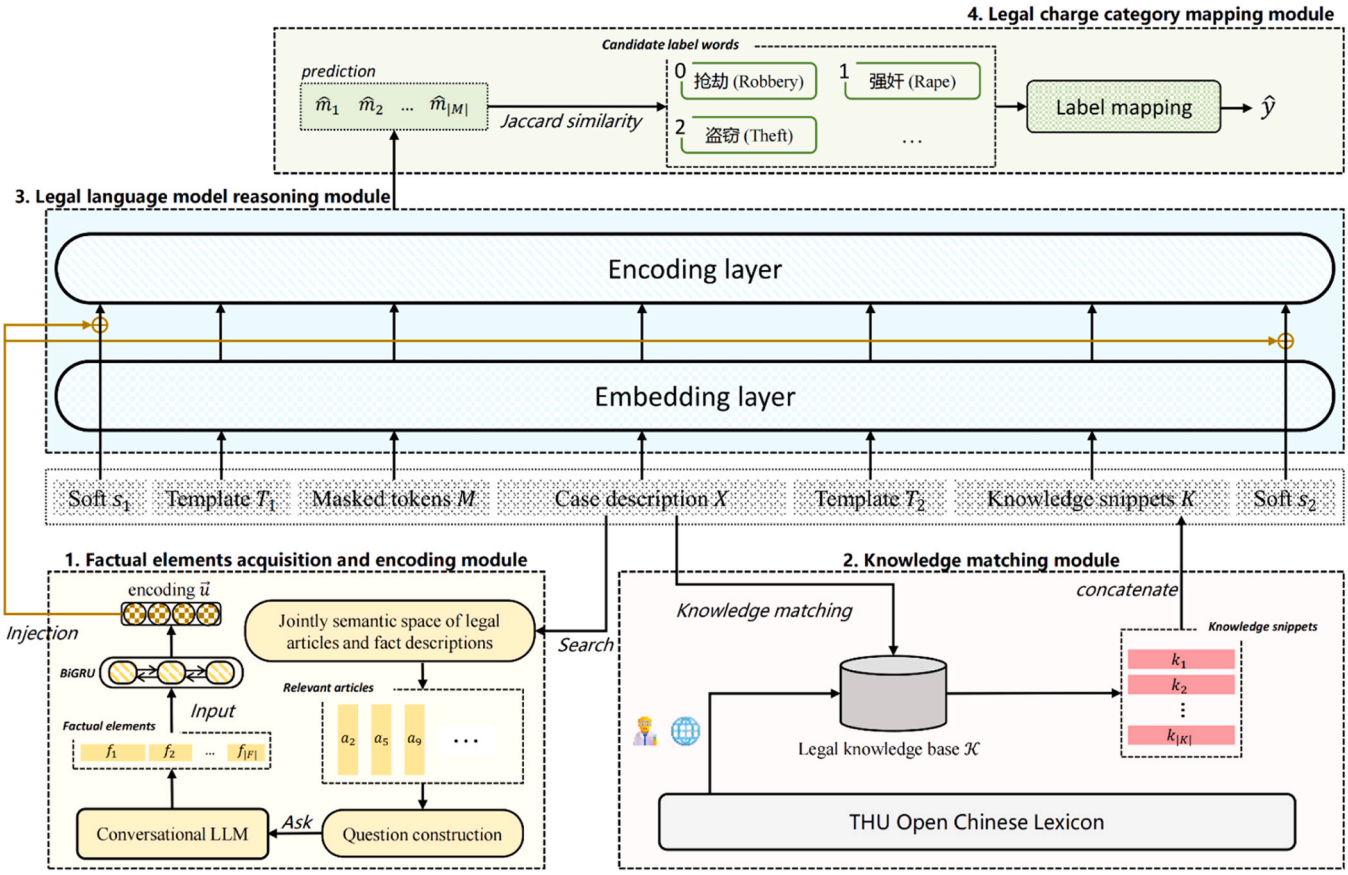

Fig. 2. Model architecture diagram. The bottom left section represents the factual element knowledge acquisition module, while the bottom right section illustrates the key knowledge snippet matching module. In the factual element knowledge acquisition module, the conversational LLM serves as the core, aiming to extract potential factual elements from case descriptions. These factual elements are then encoded into a semantic vector using a BiGRU. in the key knowledge snippet matching module, the legal knowledge base is the core component, designed to match key knowledge snippets based on the case description. The middle part of the figure illustrates the inference process of the model, which consists of two layers: the embedding layer and the encoding layer.

Silveira et al. [42,43] applied BERT-based models for topic modeling of legal texts, significantly improving the model's semantic representation capabilities. Farhadishad et al. [44] combined multiple neural network structures, including BERT, LSTM, and CNN, to extract multi-layered semantic features from legal texts. Moreover, Nigam et al. [45] employed various Transformer models to model the attention mechanism on key parts of legal texts, achieving more precise topic extraction.

In recent years, knowledge graphs have provided new directions for research in topic modeling [46]. Yao et al. [47] integrated knowledge graph embeddings with topic modeling, proposing a knowledge-based topic model. This approach significantly enhances the semantic coherence of topics by combining LDA with entity vector embeddings from knowledge graphs, allowing for better topic representations of documents. Li et al. [48] developed a model based on the Hierarchical Dirichlet Process (HDP), which further improves the interpretability of topics by leveraging information from knowledge graphs. Additionally, Zou et al. [49] successfully extracted latent disease topics from electronic health records data by learning embeddings from a medical knowledge graph, demonstrating the broad application potential of knowledge graphs in topic modeling.

2.4. Research gap

Although existing studies have explored various methods for legal charge prediction, and many of these methods focus on extracting deep knowledge from case descriptions to aid reasoning, they lack the simultaneous utilization of multi-source heterogeneous knowledge from both internal and external sources. Specifically, current approaches do not leverage both the factual elements embedded in case descriptions and the knowledge fragments from external legal knowledge bases concurrently. Moreover, many factual elements are not textual segments within the case elements and therefore cannot be handled by extractive models, necessitating the use of generative models that generate content after comprehending the case. Consequently, we propose a legal charge prediction method that can simultaneously utilize multi-source heterogeneous legal knowledge. Additionally, we employ a conversational LLM to assist in the extraction of factual elements.

3. Task formalization

The task of legal charge prediction aims to accomplish the following process: given a case description X containing L tokens, the model predict a legal charge label $\hat{y}$ based on the content of X. This process can be

denoted as Eq. (1).

$$\hat{y} = Model(X;\theta) \tag{1}$$

Where θ represents the learnable parameters within the model.

In this study, we additionally utilize a set $\mathscr{A}$ of legal articles, a conversational large language model $LLM(\cdot)$, and a legal knowledge base $\mathscr{K}$ as aids, thereby resulting in the Eq. (2).

$$\hat{y} = Model(X, \mathscr{A}, LLM(\cdot), \mathscr{K}\theta) \tag{2}$$

The following section will describe the process by which the model $Model(;\theta)$ computes the legal charge prediction outcome $\hat{y}$, given the conditions of $X, \mathscr{A}, LLM(\cdot), \mathscr{K}$.

4. Methodology

Fig. 2 illustrates our method, comprising four modules. ***The Factual elements acquisition and encoding module***, as depicted in the lower-left corner of the figure, focuses on acquiring and encoding factual elements from the given case description X. In this module, case description X search for the most relevant N legal articles via a joint semantic space, subsequently consulting a conversational Large Language Model (LLM) for factual elements in the case description based on these legal articles. The factual elements aquired are encoded into a semantic vector $\vec{u}$ by a BiGRU encoder, and $\vec{u}$ is then injected into the forward computation of the soft prompt tokens to enhance the model's reasoning capabilities for the task. ***The Knowledge matching module*** involves knowledge matching based on a legal knowledge base, wherein knowledge from the case description X is matched with a knowledge base $\mathscr{K}$. The knowledge snippets matched are then concatenated and added to the original input as prompt to further strengthen the model's reasoning. ***The Legal language model reasoning module*** is the cornerstone of our method consisting of a legal language model, as illustrated in the central part of the figure. The input to this legal language model includes five components: 1) two soft prompt tokens s_1 and s_2; two manually constructed template texts T_1 and T_2; 3) the masked tokens M; 4) the case description X; and 5) the knowledge snippets K. The objective of the legal language model is to predict the tokens at the masked positions. ***The Legal charge category mapping module*** aims to map the predictions of the language model at the masked positions onto a legal charge category $\hat{y}$, serving as the final process of our method.

In the following sections, we detail each module: Section 4.1 covers the first module, Section 4.2 the second, Section 4.3 the third, and Section 4.4 the fourth.

4.1. Factual elements acquisition and encoding

This section introduces the first module of our method, which involves acquiring and encoding factual elements from the given case description. Factual elements in case descriptions are crucial for legal judgments, as they influence the overall understanding of the cases and the final verdicts [50]. These elements typically include the time and place of the event, the individuals involved, and the specific process of the event. Example 1 demonstrates a case description and its contained factual elements.

In our method, the acquisition and encoding of factual elements involve the following four processes. Firstly, we utilize the case descriptions and legal articles from the entire training set to learn a joint semantic space. Then, when a case description X is given, this joint semantic space is employed to find the N legal articles most relevant to it. Subsequently, these N legal articles are used to consult a conversational LLM about the most noteworthy factual elements in the case. Finally, the obtained factual elements are encoded into a semantic vector $\vec{u}$. The following sections will elaborate on these processes.

Case description:

柯某某在未经著作权人授权的情况下，采用“火车采集器”网络爬虫软件，从视频网站采集5万余部电影，存储在租用的服务器上。柯某某将存储在服务器的影视作品转载到其个人运营管理的网站上提供给网民免费观看，同时收取广告费，非法获利共计人民币35万余元 (Without the authorization of the copyright owners, Ke employed “Train Collector” web crawling software to collect over 50,000 movies from a video website, storing them on a rented server. Ke then reposted these movies on a website personally operated and managed, offering them for free viewing to netizens. Concurrently, he collected advertising fees, illegally profiting a total of over 350,000 yuan)

Factual elements:

- Subject: Ke
- Authorization: Ke acted without the authorization of the copyright holders.
- Tool used: Utilized “Train Collector” web crawling software.
- Storage method: Stored the works on a rented server.
- Profit method: Earned revenue through advertising fees.
- Illicit Gains: Illegally profited a total of over 350,000 yuan.

Example 1. A case description and its contained factual elements

4.1.1. Jointly semantic space learning

We manage to utilize relevant legal articles to extract noteworthy factual elements from case descriptions, as these articles explicitly define which factual elements are pertinent to specific legal charges. For instance, the legal article pertinent to the crime of copyright infringement is: “*Acts such as copying and distributing literary, audio-visual, and computer software works for profit without the permission of the copyright holder, publishing books that are subject to another's exclusive publishing rights without consent, duplicating audio-visual products created by others without their permission, producing and exhibiting art works falsely attributed to another, where the amount of illegal gains is substantial or other serious circumstances are present*”. From this, we can infer that elements such as whether the intent was for profit, whether copyright permission was obtained, and the amount of illegal gains, are factual aspects worthy of attention.

To match relevant legal articles with case descriptions, we propose the construction of a joint semantic space of both case descriptions and legal articles. We engage in contrastive training of the language model RoBERTa [34] to facilitate its learning of this joint semantic space. Compared to rule-based methods [51,52], the language model can model deeper semantic connections between case descriptions and legal articles, thereby achieving superior matching outcomes. Furthermore, contrastive training places greater emphasis on the relative relationship between positive and negative samples compared to traditional neural network-based semantic matching methods [53]. Consequently, contrastive training aids RoBERTa in learning more distinct and discriminative features which are crucial in determining the relevance between case descriptions and legal articles. Next, we introduce the specific steps of using RoBERTa to learn the joint semantic space.

Step 1: Construct positive and negative pairs

Each case description in CAIL-2018, the largest dataset of legal charge prediction, has been labeled its relevant legal articles. Therefore, we can easily construct contrastive positive and negative pairs from the entire training set automatically. Given the training set containing the pairs of case descriptions and relevant legal articles $\mathscr{D}_{Train} = \{(X_1, \mathscr{R}_1),$

$(X_2, \mathscr{R}_2), (X_3, \mathscr{R}_3), \ldots\}$, we use Algorithm 1 to automatically construct a set $\mathscr{P}$ of positive and negative pairs. Where, X_i represents the i_{th} case description in the training set, and $\mathscr{R}_i$ represents the legal articles related to it.

Algorithm 1. Construction of the positive and negative pair set

Input: Training set $\mathcal{D}_{Train} = \{(X_1, \mathcal{R}_1), (X_2, \mathcal{R}_2), (X_3, \mathcal{R}_3), \ldots\}$

Output: Set $\mathcal{P}$ of positive and negative pairs

Initialize an empty set $\mathcal{P} = \{\}$
for each $(X_i, \mathcal{R}_i)$ in $\mathcal{D}_{Train}$ **do**
  $C \leftarrow 0$ //Count the number of positive pairs
  for each legal article r in $\mathcal{R}_i$ **do**
    Add (X_i, r) into $\mathcal{P}$ //(X_i, r) is a positive pair
    $C \leftarrow C + 1$
  end for
  Randomly select C number of $\mathcal{R}^{neg}$ from $\mathbb{C}_{\mathcal{D}_{Train}}(X_i, \mathcal{R}_i)$
  Select a legal article r^{neg} from each $\mathcal{R}^{neg}$
  Pair X_i with each legal article r^{neg}: (X_i, r^{neg})
  Add C number of (X_i, r^{neg}) into $\mathcal{P}$ //(X_i, r^{neg}) is a negative pair
end for
return $\mathcal{P}$

Step 2: Obtain representations of case descriptions and legal articles

We have obtained set $\mathscr{P}$ of positive and negative pairs in the first step. Each pair in the set is either a related pair of case description and legal article or is irrelevant. In this step, we employ RoBERTa to acquire semantic vectors for the case description and legal article in each pair. This process is illustrated in Eq. (3).

$$\boldsymbol{P} = RoBERTa(\mathscr{P}) \tag{3}$$

Herein, $\boldsymbol{P}$ represents a matrix, where the odd-numbered columns of $\boldsymbol{P}$ denote the semantic vectors of case descriptions, and the even-numbered columns represent the semantic vectors of legal articles. Now that we have obtained the semantic vectors for all samples in positive and negative pairs, we proceed to train RoBERTa using a contrastive loss.

Step 3: Train the RoBERTa via contrastive loss

During the training, RoBERTa learns to decrease the semantic distance between samples in positive pairs while increasing the distance between those in negative pairs. This objective is achieved through a contrastive loss function, which quantifies the similarity between semantic vectors in a pair. The calculation of the loss for the i_{th} case description is as shown in Eq. (4).

$$\ell_i = -\log \frac{\sum_{c=1}^{C} e^{sim(\vec{p}_i, \vec{p}^{+}_c)/\tau}}{\sum_{c=1}^{C} \left(e^{sim(\vec{p}_i, \vec{p}^{+}_c)/\tau} + e^{sim(\vec{p}_i, \vec{p}^{-}_c)/\tau}\right)} \tag{4}$$

Where, $\vec{p}_i$ represents the semantic vector of the i_{th} case description, while $\vec{p}^{+}_c$ and $\vec{p}^{-}_c$ respectively denote the semantic vectors of the legal articles in the c_{th} positive and negative pairs. Besides, $sim(\vec{p}_1, \vec{p}_2)$ denotes the cosine similarity between vectors $\vec{p}_1$ and $\vec{p}_2$, and τ is a temperature hyperparameter.

The trained RoBERTa model can encode case descriptions and legal articles into a joint semantic space, where the representation of a case description and its corresponding legal articles exhibit a closer semantic distance within this space.

4.1.2. Relevant legal articles searching

Having obtained a joint semantic space for case descriptions and legal articles through prior operations, this section utilizes this joint semantic space to search N legal articles relevant to a given case description. Given the case description X and a set of candidate legal articles $\mathscr{A}$, we first encode them into the joint semantic space using the trained RoBERTa model, obtaining their respective semantic vectors. This process is illustrated in Eq. (5).

$$\vec{x}, [\vec{a}_1, \vec{a}_2, \ldots, \vec{a}_{\mathscr{A}}] = RoBERTa(X, \mathscr{A}) \tag{5}$$

Where, $\vec{x}$ represents the semantic vector of case description X, and $\vec{a}_i$ denotes the semantic vector of the i_{th} legal article in set $\mathscr{A}$.

Subsequently, we compute the relevance between the case description and each candidate legal article through vector inner product, as shown in Eq. (6), thereby selecting the N legal articles with the highest relevance.

$$Relevance(\vec{x}; \vec{a}_i) = \vec{x} \cdot \vec{a}_i \tag{6}$$

In the next section, we will utilize the N legal articles searched, along with case description X, to consult a conversational large language model. This is done to acquiring noteworthy factual elements within X.

4.1.3. Conversational large language model consultation

Conversational LLMs, with their vast parameter count, possess robust

contextual reasoning capabilities and have learned a wealth of generic world knowledge during their pre-training. Furthermore, Conversational LLMs often exhibit strong zero-shot reasoning capabilities, thus enabling their direct use as ready-made tools without the need for additional fine-tuning. Based on these reasons, we use a Conversational LLM to assist us in acquiring factual elements from X. We construct the following question template.

We utilize this template to conduct inquiries with the conversational LLM, resulting in a list of factual elements denoted as $F = [f_1, f_2, \ldots, f_{|F|}]$. These factual elements are subsequently be encoded as semantic vectors in Section 4.1.4. When the dataset is in Chinese, we utilize GLM-130B, a Chinese conversational LLM developed by ZhiTu HuaZhang Technology Co., Ltd., in this process. When the dataset is in English, we directly invoke the API of ChatGPT to achieve this process.

Question template

Messeges=
[{"role": "user", "content": "Factual elements in a case description refer to: specific facts used to describe and prove the circumstances of the case, including basic information such as time, location, characters, and the sequence of events."},
{"role": "assistant", "content": "That's correct. Factual elements in a case description are indeed the specific details used to outline and substantiate the circumstances of a case."},
{"role": "user", "content": "Please analyze the case description in < > based on the legal articles in < < > >, and list 5–10 factual elements into []"}]

4.1.4. Factual elements encoding

Given the list of factual elements, $F = [f_1, f_2, \ldots f_{|F|}]$, obtained from the previous process, we concatenate them and input the combined sequence into a BiGRU encoder. A BiGRU consists of two GRU layers that process the data in opposite directions: one forward GRU and one backward GRU. The forward GRU processes the sequence from f_1 to $f_{|F|}$, and the backward GRU processes it from $f_{|F|}$ to f_1. Each GRU updates its hidden state at each step in the sequence.

Let $\overrightarrow{h}_t$ be the hidden state of the forward GRU at time step t, and $\overleftarrow{h}_t$ be the hidden state of the backward GRU at time step t. They are computed as Eqs. (7–8).

$$\overrightarrow{h}_t = GRU(f_t, \overrightarrow{h}_{t-1}) \tag{7}$$

$$\overleftarrow{h}_t = GRU(f_t, \overleftarrow{h}_{t+1}) \tag{8}$$

The final semantic vector $\overrightarrow{u}$ is typically obtained by concatenating the last hidden state of the forward GRU and the first hidden state of the backward GRU, as shown in Eq. (9).

$$\overrightarrow{u} = [\overrightarrow{h}_{|F|}; \overleftarrow{h}_1] \tag{9}$$

The semantic vector, obtained through this process, encapsulates the information of all factual elements derived from the case description. This vector will subsequently be integrated into the inference model to enhance the model's prediction capabilities regarding legal charges.

4.2. Knowledge matching

This section introduces the second module of our method. This module matches case description X with a given knowledge base $\mathscr{K}$. The knowledge snippets matched serve as prompts to enhance the reasoning model's prediction of legal charges.

We use *THUOCL_Law* as the 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 knowledge in *THUOCL_Law*.

As presented in the table, a knowledge snippet is essentially a keyword. As [50] discussed, keywords in case descriptions are crucial for predictions of legal charges. We simply utilize regular expressions to match these keywords from the case description X, thereby obtaining a list of keywords $K = [k_1, k_2, \ldots, k_{|K|}]$, also referred to as the list of knowledge snippets.

The concatenation of the knowledge snippets in K will serve as a prompt, and in conjunction with other components, act as the input for the inference model.

4.3. Legal language model reasoning

This section introduces the third module of our method. In this module, we employ a legal language model to reason the legal charge associated with the given case description X. Traditionally, the task of predicting legal charges is viewed as a classification problem, where the model's output is directly a probability distribution, and the index of the highest probability is the predicted label. However, we transform the task of legal charge prediction into a language modeling (cloze test) task, prompting the model to predict masked tokens. Then, we map the predictions at these masked positions onto the final category labels.

To implement the language modeling task, we construct hard prompt templates T_1 and T_2, as follows:

T_1 = "*He will be charged with criminal responsibility for*"

T_2 = "*Keywords in the case description are as follows:*"

These hard prompts serve as part of the input for the inference model, guiding the model to predict masked tokens. In addition to hard prompts, we also incorporate two soft prompts s_1 and s_2 into the input. Semantic vector $\overrightarrow{u}$ obtained in Section 4.1 will be merged with these soft prompts, injecting the knowledge about factual elements into the model's inference. Moreover, the masked tokens $M = [m_1, m_2, \ldots, m_{|M|}]$ and case description $X = [x_1, x_2, \ldots, x_L]$ are also essential components of the input. Lastly, the concatenation of knowledge snippets K, acquired in Section 4.2, is also included as a part of the input.

In summary, the input for the inference model is composed of the case description X, hard prompt texts T_1 and T_2, soft prompts s_1 and s_2, the masked sequence $M = [m_1, m_2, \ldots, m_{|M|}]$, and the concatenation of knowledge snippets K, as illustrated in Eq. (10).

$$\begin{aligned} X' = [&s_1, t_{1,1}, t_{1,2}, \ldots, t_{1,|T_1|}, m_1, m_2, \ldots, m_{|M|}, x_1, x_2, \ldots x_L, t_{2,1}, t_{2,2}, \\ &\ldots, t_{2,|T_2|}, k_1, k_2, \ldots, k_{|K|}, s_2] \end{aligned} \tag{10}$$

Where, $t_{1,i}$ represents the i_{th} token in T_1, and $t_{2,i}$ denotes the i_{th} token in T_2. Besides, m_i stands for the i_{th} mask, x_i refers to the i_{th} token in X, and k_i indicates the i_{th} token in K.

To facilitate understanding, we illustrate X' more intuitively through the example provided in Fig. 3. In the diagram, two purple tokens

Table 1
Part of the knowledge 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)

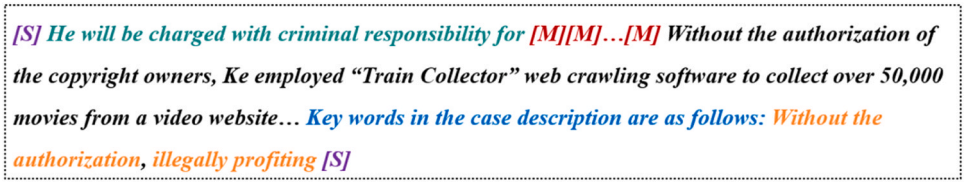

Fig. 3. Schematic diagram of the components of X'.

represent the soft prompts, the green section is T_1 and the blue section is T_2. The red tokens indicate the masked tokens, the black text is the original case description X, and the yellow section is the concatenation of keywords.

Nextly, we input X' into the inference model, which is a pre-trained legal language model. The inference model consists of embedding and encoding layers, as shown in Fig. 1. During the embedding layer stage, all tokens in X' except the soft prompts are embedded by the embedding layer of the inference model. At the same time, the soft prompts in X' are embedded by an additional trainable embedding matrix. This process is shown in Eq. (11).

$$\overrightarrow{e}_i = \begin{cases} \boldsymbol{S}[i], if\, i \in soft_{idx} \\ Embedding(token_i), otherwise \end{cases} \tag{11}$$

Where $\boldsymbol{S} \in \mathbb{R}^{|soft_{idx}| \times d_h}$ is a trainable embedding matrix, and $soft_{idx}$ is the index of the soft prompt token. d_h is embedding dimension of the model. Therefore, an embedding vector sequence $\boldsymbol{E}$ of all the tokens (including soft prompt tokens) in X' can be obtained by the equation.

$$\boldsymbol{E} = [\overrightarrow{e}_{s_1}, \overrightarrow{e}_{t_{1,1}}, \overrightarrow{e}_{t_{1,2}}, \ldots, \overrightarrow{e}_{t_{1,|T_1|}}, \overrightarrow{e}_{m_1}, \overrightarrow{e}_{m_2}, \ldots, \overrightarrow{e}_{m_{|M|}}, \overrightarrow{e}_{x_1}, \overrightarrow{e}_{x_2}, \ldots, \overrightarrow{e}_{x_L}, \overrightarrow{e}_{t_{2,1}}, \ldots, \overrightarrow{e}_{t_{2,|T_2|}}, \overrightarrow{e}_{k_1}, \overrightarrow{e}_{k_2}, \ldots, \overrightarrow{e}_{k_{|K|}}, \overrightarrow{e}_{s_2}]$$

Where $\overrightarrow{e}_{t_{1,i}}$, $\overrightarrow{e}_{t_{2,i}}$, $\overrightarrow{e}_{m_i}$, $\overrightarrow{e}_{x_i}$ and $\overrightarrow{e}_{k_i}$ denote the embedding vectors of the i_{th} tokens in T_1, T_2, M, X, and K respectively. Besides, $\overrightarrow{e}_{s_1}$ and $\overrightarrow{e}_{s_2}$ denote the embedding vectors of the soft prompt tokens s_1 and s_2.

To inject the inference model with factual element information during its forward computation, we add the semantic vector $\overrightarrow{u}$ obtained in Section 4.1 to the vectors $\overrightarrow{e}_{s_1}$ and $\overrightarrow{e}_{s_2}$ respectively. This results in prompt vectors enriched with factual element information. This process is demonstrated in Eqs. (12) and (13).

$$\overrightarrow{e}'_{s_1} = \overrightarrow{e}_{s_1} + \overrightarrow{u} \tag{12}$$

$$\overrightarrow{e}'_{s_2} = \overrightarrow{e}_{s_2} + \overrightarrow{u} \tag{13}$$

Subsequently, we replace $\overrightarrow{e}_{s_1}$ and $\overrightarrow{e}_{s_2}$ in $\boldsymbol{E}$ using $\overrightarrow{e}'_{s_1}$ and $\overrightarrow{e}'_{s_2}$ to obtain:

$$\boldsymbol{E}' = [\overrightarrow{e}'_{s_1}, \overrightarrow{e}_{t_{1,1}}, \overrightarrow{e}_{t_{1,2}}, \ldots, \overrightarrow{e}_{t_{1,|T_1|}}, \overrightarrow{e}_{m_1}, \overrightarrow{e}_{m_2}, \ldots, \overrightarrow{e}_{m_{|M|}}, \overrightarrow{e}_{x_1}, \overrightarrow{e}_{x_2}, \ldots, \overrightarrow{e}_{x_L}, \overrightarrow{e}_{t_{2,1}}, \ldots, \overrightarrow{e}_{t_{2,|T_2|}}, \overrightarrow{e}_{k_1}, \overrightarrow{e}_{k_2}, \ldots, \overrightarrow{e}_{k_{|K|}}, \overrightarrow{e}'_{s_2}]$$

Finally, we input $\boldsymbol{E}'$ into the encoding layer of the inference model and obtain the hidden layer outputs of the model, that is, the contextual representations for each token. This process is shown as Eq. (14).

$$\boldsymbol{R} = \overrightarrow{r}_{s_1}, \overrightarrow{r}_{t_{1,1}}, \overrightarrow{r}_{t_{1,2}}, \ldots, \overrightarrow{r}_{t_{1,|T_1|}}, \overrightarrow{r}_{m_1}, \overrightarrow{r}_{m_2}, \ldots, \overrightarrow{r}_{m_{|M|}}, \overrightarrow{r}_{x_1}, \overrightarrow{r}_{x_2}, \ldots, \overrightarrow{r}_{x_L}, \overrightarrow{r}_{t_{2,1}}, \ldots, \overrightarrow{r}_{t_{2,|T_2|}}, \overrightarrow{r}_{k_1}, \overrightarrow{r}_{k_2}, \ldots, \overrightarrow{r}_{k_{|K|}}, \overrightarrow{r}_{s_2} = Encoding(\boldsymbol{E}') \tag{14}$$

The model's objective is to predict the tokens at the masked position. Therefore, by projecting $\overrightarrow{r}_{m_1}, \overrightarrow{r}_{m_2}, \ldots, \overrightarrow{r}_{m_{|M|}}$ into the vocabulary space, we can obtain the probability distributions of the predicted tokens. Subsequently, by selecting the indices with the highest probabilities, the model can determine the tokens at the masked positions. We denote the i_{th} predicted token at the masked positions as $\widehat{m}_i$.

The next section describe the process of mapping predicted tokens $\widehat{m}_1, \widehat{m}_2, \ldots, \widehat{m}_{|M|}$ to a charge category label.

4.4. Legal charge category mapping

We need to map the outcomes of the language modeling (cloze test) task back to the classification task. To this end, this section constructs a mapping from predicted tokens to legal charge categories. We calculate the Jaccard similarities between the predicted tokens and the texts of the legal charge labels. For instance, if the predicted label text has the highest Jaccard similarity with "Manufacture, sell, and disseminate obscene materials", then the legal charge predicted by the model for the current case description is "Manufacture, sell, and disseminate obscene materials". Table 2 shows some of the texts of the legal charge labels.

Therefore, the final prediction of the model is shown in Eq. (15).

$$\widehat{y} = argmax(Jaccard(\widehat{m}_{1:|M|}, v_y)) \tag{15}$$

Where $\widehat{y}$ is the final label predicted by the model, and $\widehat{m}_{1:|M|}$ denotes the predicted tokens at the masked positions, and v_y denotes the text of category y.

5. Experimental settings

This section details the experimental setup, including the datasets used, baselines, hyperparameters used and evaluation metrics of our method.

Table 2
Some of the label texts.

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)

Table 3
Some instances in the CAIL2018 dataset.

Case description	Legal charge 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…，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 of the three datasets. The first column represents the number of training samples, the second column indicates the number of development samples, the third column shows the number of test samples, and the last column provides the total number of samples.

	Train	Dev	Test	Total
CAIL	1,605,645	635,215	635,215	2,676,075
CJO	604,646	201,548	201,548	1,007,744
PKL	105,446	35,148	35,148	175,744

5.1. Datasets

We utilized CAIL-2018 [9], the largest Chinese legal charge prediction dataset, as our experimental dataset. The authors acquired 2.6 million public criminal cases from the Supreme People's Courts of China and subjected them to preprocessing, ultimately obtaining 2,676,075 case description texts accompanied by 196 unique legal charge labels. Each case description corresponds to only one legal charge label, so the task is a single-label classification problem. Table 3 shows some instances from the CAIL2018 dataset. In addition, 3/5 of the total are used as the training set, 1/3 as the development set, and the remaining 1/3 as the test set.

Additionally, we utilized the CJO and PKL datasets constructed by Zhong et al. [54] for our experiments. The CJO dataset was constructed by Zhong et al. based on criminal cases published by the Chinese government on the China Judgments Online platform. This dataset includes one million cases and 99 candidate charges. We divide the dataset into three parts: three-fifths is allocated as the training set, one-fifth as the development set, and the remaining one-fifth as the test set. The PKU dataset, also constructed by Zhong et al., comprises 170,000 criminal cases published by Peking University Law Online and includes 64 candidate charge labels. We similarly divided this dataset into training, development, and test sets using a ratio of 3/5, 1/5, and 1/5, respectively. Table 4 presents the statistical information of the three datasets.

5.2. Baselines

We compare the proposed method against the following baselines to verify its advancement.

(1) **CNN** [1]: The method proposed by Wang et al., which uses convolutional neural networks to model the terminologies and keywords within case descriptions.
(2) **BiLSTM** [2]: The method proposed by Yang et al., which models the global semantics of case descriptions via bidirectional long and short-term memory networks.
(3) **BiGRU** [3]: The method proposed by Chen et al. Compared to bidirectional long and short-term memory networks, bidirectional gated networks have lower computational complexity for modeling the global semantics of case descriptions.
(4) **Attention** [5]: The attention mechanism-based method proposed by Sukanya and Priyadarshini. Attention mechanisms can assign different weights to different factual information in a case description. This method achieved state-of-the-art results across multiple benchmarks.
(5) **BERT** [50]: The method of fine-tuning a BERT on the legal charge prediction task. This is a powerful baseline.
(6) **HMN** [20]: This is a hierarchical matching network for crime classification proposed by Wang et al. This method is a novel and strong baseline on the CAIL2018 dataset.
(7) **ChatGLM**: We directly engage in dialog with the Chinese conversational LLM, ChatGLM, to obtain the legal charge label corresponding to each case description.

5.3. Hyperparameters used

All our experiments were conducted on a 40 G A100 GPU. During the training phase, the model employed a learning rate of 1e-5, a batch size of 8. We employ Lawformer [25], a recently proposed legal pre-trained language model, as our inference model, which can accept text inputs up to a maximum length of 4000. The number of masked tokens to be predicted is set to 20. Besides, we set a maximum training epoch of 50 with an early stopping mechanism. We use AdamW as the optimizer.

Additionally, the other parameter settings in our method are as follows: the maximum retrieval number of related articles in the factual element extraction module is set to 10; the hidden layer dimension of the BiGRU is set to 500; the maximum number of knowledge fragment

Table 5
Hyperparameter settings for baseline methods. "LR" Denotes the learning rate.

	Filters	Filter Size	Hidden Units	Heads	Optimizer	LR	Batch Size	Epochs
CNN	64	5	/	/	AdamW	1e−4	32	50
BiLSTM	/	/	256	/	AdamW	1e−4	32	50
BiGRU	/	/	256	/	AdamW	1e−4	32	50
Attention	/	/	/	16	AdamW	1e−4	32	50
BERT	/	/	/	/	AdamW	1e−4	8	20
HMN	/	/	/	16	AdamW	1e−4	32	50
ChatGLM	/	/	/	/	/	/	/	/

Table 6
Performance of the baselines and our method.

Method	CAIL			CJO			PKL		
	P	R	F1	P	R	F1	P	R	F1
CNN	0.72	0.70	0.71	0.41	0.32	0.36	0.62	0.50	0.55
BiLSTM	0.69	0.67	0.68	0.39	0.30	0.34	0.58	0.40	0.47
BiGRU	0.66	0.66	0.66	0.44	0.34	0.38	0.57	0.43	0.49
Attention	0.77	0.76	0.76	0.41	0.31	0.35	0.63	0.62	0.62
BERT	0.62	0.61	0.61	0.50	0.39	0.44	0.55	0.38	0.45
HMN	0.79	0.77	0.78	0.51	0.37	0.43	0.62	0.52	0.57
ChatGLM	0.69	0.68	0.68	0.50	0.38	0.43	0.53	0.40	0.46
Ours	**0.85**	**0.83**	**0.84**	**0.59**	**0.55**	**0.57**	**0.69**	**0.60**	**0.64**

matches is set to 10; and the dimension of the soft prompt vector is aligned with the embedding layer dimension of Lawformer. The hyperparameters for the baseline methods are set according to their original paper, as detailed in Table 5.

5.4. Evaluation metrics

We use three evaluation metrics: Precision (P), Recall (R), and F1-score. Precision, also known as the positive predictive value, measures the proportion of correctly predicted positive instances out of all instances predicted as positive. Recall, also known as sensitivity or the true positive rate, measures the proportion of actual positive instances that were correctly identified by the model. The F1 score is the harmonic mean of Precision and Recall, providing a single metric that balances both. These three evaluation metrics are widely utilized in legal charge prediction tasks.

6. Results and discussion

This section discusses the experimental results. Section 6.1 compares our method with baselines, while Section 6.2 validates the effectiveness of each module within our method through ablation experiments. Moreover, Section 6.3 analyzes the impact of the training data size on the performance of our method. Section 6.4 analyzes the hyperparameter settings, and Section 6.5 validates the interpretability of our model through a specific case study.

6.1. Comparison with baselines

Table 6 displays the performance of our method compared to the baselines. From the table, it can be observed that the performances of ***CNN***, ***BiLSTM***, and ***BiGRU*** are relatively similar, with macro F1 scores ranging between 0.66 and 0.71. ***CNN*** performs the best, which may be attributed to the fact that case descriptions are often lengthy, and thus modeling the sequence structure is less effective than modeling key information. Among the traditional neural network models presented in the first four rows, ***Attention*** achieves the best performance. Attention mechanisms are capable of focusing on local key information within case descriptions as well as the correlations between different pieces of information, hence achieving performance significantly beyond that of other traditional neural network baselines. Based on this, we can also infer that factual elements and knowledge snippets within case descriptions can enhance the effectiveness of legal charge prediction.

From the fifth row of the table, it is evident that the usually strong baseline ***BERT*** performs worse than traditional neural network models, achieving only a macro F1 score of 0.61. This is due to BERT's input length limitation of 512, which prevents it from fully modeling case description texts that average over 1000 in length. In contrast, traditional neural network models do not have an input length restriction, enabling them to model case descriptions more completely and thereby achieve better results than ***BERT***. Our method employs Lawformer as the inference model, which can accept case description inputs of over 4000 in length, thereby enabling more complete modeling of case descriptions.

Table 7
Results of the ablation experiments.

	P	R	F1
Ours	0.85	0.83	0.84
– knowledge snippets	0.82	0.81	0.82 (−0.02)
– factual elements	0.81	0.81	0.81 (−0.03)
– knowledge snippets and factual elements	0.79	0.77	0.78 (−0.06)
– Contrastive training	0.83	0.82	0.82 (−0.02)

From the sixth row of the table, it is apparent that the performance of ***HMN*** surpasses other baselines due to its integration of hierarchical matching, which not only allows for the complete modeling of the entire case description but also enables hierarchical modeling of key information. Furthermore, from the seventh row of the table, we can observe that the performance of ***ChatGLM*** is mediocre. Despite ChatGLM having learned a vast amount of general semantic knowledge and possessing good zero-shot reasoning capabilities, it still exhibits hallucination issues in specialized fields such as law and medicine.

Finally, from the last row of the table, we can see that our method achieved the best results, with a macro F1 score reaching 0.84. This demonstrates that our method is feasible and effective. The effectiveness of our method compared to traditional neural network methods such as CNN, BiLSTM, BiGRU, and Attention is primarily due to our use of a pre-trained language model with 8 M parameters, which offers superior understanding of case texts. Additionally, our approach maintains an advantage over the pre-trained language model BERT for two key reasons: firstly, Lawformer allows for longer input sequences, enabling complete modeling of the entire case text in a single pass, thereby minimizing information loss; secondly, Lawformer is pre-trained on a large-scale Chinese legal corpus, endowing it with a certain degree of legal prior knowledge. Finally, even when compared to the advanced generative language model ChatGLM, our method still demonstrates significant advantages, mainly because it integrates heterogeneous legal knowledge from multiple sources during model inference, which enhances the model's ability to analyze cases. To further validate the contribution of each module in our method, we conducted ablation experiments in Section 4.2.

6.2. Ablation experiments

This section analyzes the contribution of each module in our method through ablation experiments, with the experimental results shown in Table 7. The first row of the table represents the performance of our original method on the CAIL-2018 dataset.

Firstly, we removed the knowledge snippets module, with the experimental results shown in the second row of the table. It can be observed that the macro F1 score of the model decreased by 0.02, indicating that focusing on factual elements within case descriptions indeed enhances the prediction effectiveness of legal charges. To thoroughly investigate the contribution of knowledge snippets to our method, we reduce the maximum number of matched knowledge

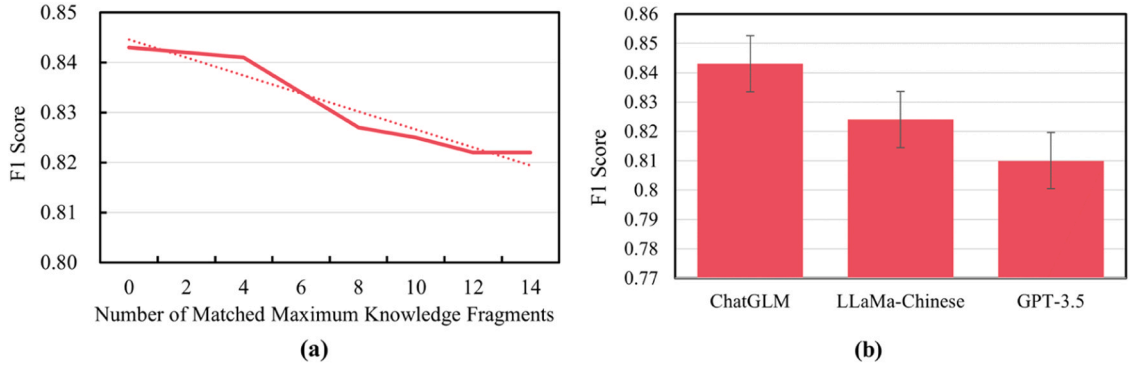

Fig. 4. Impact of knowledge snippets and factual elements on our method.

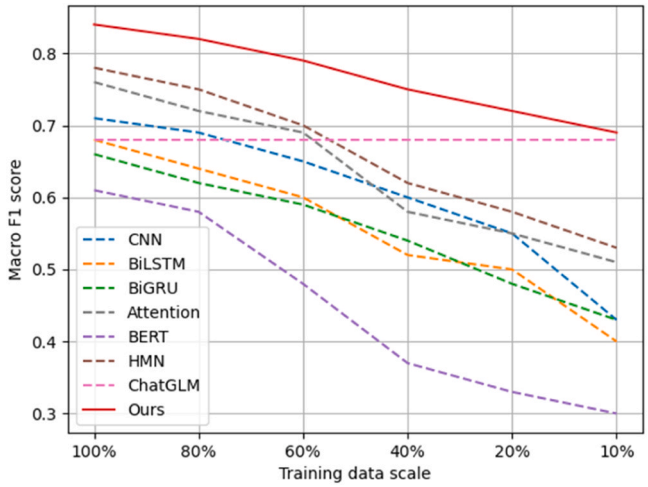

Fig. 5. Variation curves of the models' F1 scores on the test set as the training data decreases.

snippets from 12 to 0. Subfigure (a) in Fig. 4 demonstrates that the F1 score of our method is approximately proportional to the number of knowledge snippets.

Nextly, we retained the knowledge snippets module and removed the factual element acquisition and encoding module. It can be seen from the third row of the table that this resulted in a decrease of 0.03 in the model's macro F1 score. This indicates that factual elements within case descriptions also contribute beneficially to the legal charge prediction task. We evaluate the impact of different conversational LLMs on our method, as conversational LLMs are crucial for obtaining case elements. Subfigure (b) in Fig. 4 indicates that using ChatGLM enables our model to achieve the highest F1 score. This is because ChatGLM is a conversational LLM specifically developed for Chinese, making it more proficient in handling Chinese tasks.

Finally, we removed both modules simultaneously, with the experimental results shown in the fourth row of the table. It can be observed that the performance of the model significantly decreased by 0.06. The significant decline indicates that the integration of multi-source heterogeneous legal knowledge is crucial for the task of legal charge prediction.

We also validated the role of contrastive training, as discussed in Section 4.1, in accurately retrieving relevant legal articles. We removed the learning of a joint semantic space and directly used RoBERTa to encode the case descriptions and legal articles. We still determined the relevance between a given case description and each legal article by calculating the inner product between their encoded vectors. The experimental results indicate that this operation led to a decrease of 0.02 in the final results of the model, as shown in the last row of the table. This demonstrates that it is necessary to train a joint semantic space for case descriptions and legal articles through contrastive learning.

6.3. *Impact of training data volume*

This section tests the effects of using training data of different scales on the model and analyzes the experimental results. Fig. 5 illustrates the variation curves of the models' F1 scores on the test set as the training data decreases. It can be observed that with the reduction of the training data volume, the macro F1 scores of all methods decline (except for ***ChatGLM***, as it does not require training data), yet the decrease in the macro F1 score of our method is more gradual. This indicates that our approach possesses a stronger advantage in scenarios of data scarcity. Furthermore, it was found that even when the training data volume was reduced to merely 10 % of its original size, our method still achieved a macro F1 score that surpassed that of ChatGLM. This can be attributed to two factors. On one hand, the CAIL-2018 dataset itself is quite large, meaning that even 10 % of the original training data volume consists of 80,000 training samples. On the other hand, this suggests that conversational LLMs still do not hold a definitive advantage in specific domains such as law and medicine.

6.4. *Hyperparameter analysis*

This section analyzes the settings of hyperparameters in our method, with the experimental results shown in Fig. 6. The subgraphs from left to

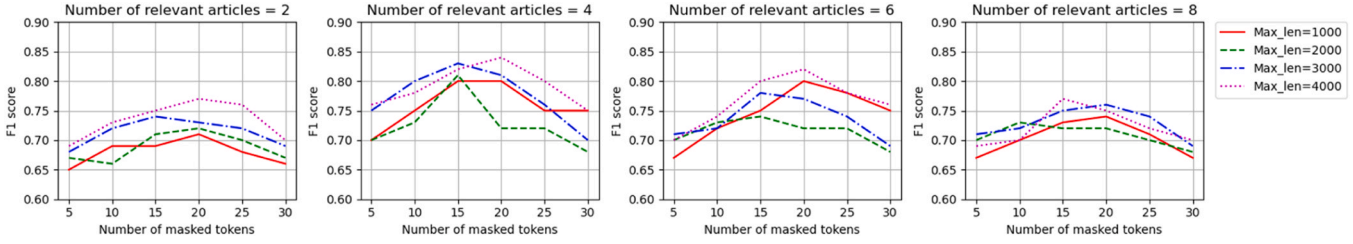

Fig. 6. Influence of hyperparameters on our model.

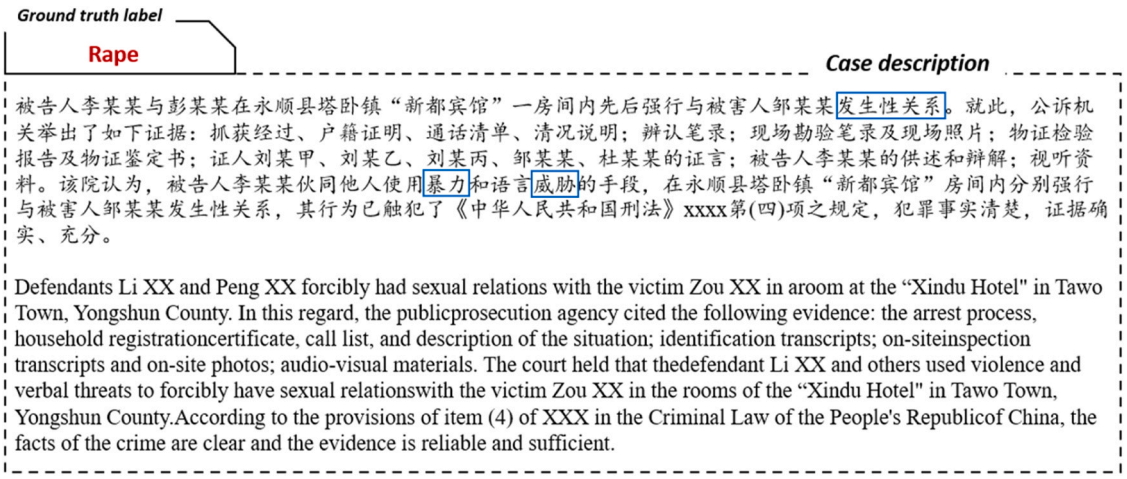

Fig. 7. Analysis of the prediction process of our method on a given case description.

right correspond to the number of relevant articles searched being 2, 4, 6, and 8, respectively. The vertical axis of each subgraph represents macro F1 score. Overall, it can be observed that the model achieves the best F1 score when the number of relevant articles searched is 4. This indicates that searching too many relevant articles may introduce noise to the conversational LLM's process of extracting factual elements from case descriptions.

Additionally, we examine each subgraph. The different colored lines within each subgraph represent the maximum truncation length of case descriptions, set at 1000, 2000, 3000, and 4000, respectively. It is evident that as the maximum truncation length of case descriptions increases, the model's performance improves. This is because the larger the maximum truncation length, the more complete the model's semantic understanding of case descriptions, thereby achieving better results. Lastly, the horizontal axis in each subgraph represents the number of masked tokens to be predicted. It can be observed from each subgraph that the model tends to achieve higher F1 values when the number of masked tokens is set to 15 and 20. Therefore, we set the number of masked tokens to be predicted by the model to 20.

6.5. Case study

In this section, we qualitatively analyze the prediction of our method for a given case description to validate the interpretability and effectiveness of the model. In Fig. 7, a case description is provided with its true label identified as “Rape.” From the figure, it is evident that our method's knowledge matching module successfully matched the following knowledge snippets from the case description: 发生性关系 (occurrence of sexual relation), 暴力(violence), and 威胁(threat). Intuitively, we can agree that these three keywords are highly related to the crime of rape. Therefore, it can be said that the knowledge matching module provides evidence for the model's prediction. This demonstrates that our proposed method has a certain degree of interpretability.

Additionally, the factual element acquisition module successfully extracted the factual elements within the case description via the conversational LLM. These factual elements include:

Fig. 8. A legal charge prediction app developed for mobile devices based on our proposed method.

- **Location**: In a room at “New Capital Hotel” in Tawo Town, Yongshun County.
- **Sexual acts**: Defendant Li and Defendant Peng forcibly engaged in sexual acts with the victim Zou in the room at “New Capital Hotel.”
- **Means used to commit the crime**: Defendant Li, along with others, used violence and verbal threats.
- **Crime time**: The specific time is not mentioned.

These factual elements also contribute to providing interpretability for legal professionals or users without legal expertise. In summary, this section demonstrates that our method exhibits a high level of interpretability, a crucial aspect in the legal domain.

Table 8
Method ranking table.

	CNN	BiLSTM	BiGRU	Attention	BERT	HMN	ChatGLM	Ours
CAIL	4	5.5	7	3	8	2	5.5	1
CJO	6	8	5	7	2	3.5	3.5	1
KPL	4	6	5	2	8	3	7	1

6.6. Friedman test

The Friedman test is a non-parametric statistical procedure designed to compare multiple treatments, models, or methods under repeated measures or matched conditions, without assuming normality or homogeneity of variances [55]. By ranking the performance of each method for every dataset, the Friedman test aggregates these rankings across all experimental units to determine whether the observed differences in median ranks are statistically significant [56].

We first measure the F1 scores of seven baseline methods and our proposed method on three datasets (CAIL, CJO, and KPL). Then, we rank the methods' performance on these datasets to obtain the ranking table shown in Table 8 and calculate each algorithm's average rank. We compute the test statistic via Eq. (16). The resulting statistic is 11.8, and the p-value is 0.1039. Since this p-value exceeds the threshold $\alpha = 0.05$, we conclude that there is no statistically significant difference in the overall average ranks of all the methods.

$$\chi_F^2 = \frac{12N}{k(k+1)}[\sum_{j=1}^{k} R_j^2 - \frac{k(k+1)^2}{4}] \tag{16}$$

6.7. Practical application

We developed a legal charge prediction app based on the proposed method, which can be operated on mobile devices, as illustrated in Fig. 8. The app comprises two main interfaces: the case description upload interface, shown on the left side of the figure, and the charge prediction and explanation interface, depicted on the right side. When uploading case descriptions, users can select one of three large models for parsing case elements: ChatGLM, LLaMa, or GPT-3.5. Additionally, users have the option to choose the knowledge base utilized, although currently, only one knowledge base, THUOCL_Law, is available. To ensure effective charge prediction, the length of the case description entered by a user must exceed 200 tokens.

The method we propose improves the accuracy of legal charge prediction tasks, thereby providing users with more reliable legal assistance services. Additionally, the intermediate results of our approach, including factual elements and knowledge snippets, enable users to understand the correlation between the prediction outcomes and the case information.

6.8. Limitations

Although we have demonstrated that leveraging multi-source heterogeneous legal knowledge can significantly improve the accuracy and interpretability of legal charge prediction, our approach still faces the some challenges. Firstly, our method is heavily dependent on the quality of the knowledge base. A low-quality legal knowledge base can have a significant adverse impact on the model. Additionally, our approach requires the use of a conversational LLM to generate case elements based on case descriptions, but conversational LLMs may sometimes produce hallucinations. To mitigate these limitations, we propose the following future research directions: (1) To avoid the negative impact of low-quality knowledge bases, multiple knowledge bases can be used concurrently, with an LLM overseeing the quality of the knowledge retrieved; (2) To alleviate the hallucinations produced by conversational LLMs in this task, fine-tuning existing LLMs using legal data could be considered.

7. Conclusion

We propose a multi-source heterogeneous knowledge-enhanced prompt learning method for legal charge prediction. We transform the legal charge prediction task from a classification problem into a masked language modeling problem, employing prompt learning for model training. Subsequently, the method extracts knowledge snippets from case descriptions via a legal knowledge base and obtains factual elements through relevant legal articles and a conversational LLM. By injecting factual elements and knowledge snippets into the model's reasoning in different ways, the model's understanding of case descriptions is enhanced. We also introduce a joint semantic space learning for case descriptions and legal articles using a contrastive loss to more accurately identify legal articles relevant to a case description. Experimental results demonstrate that our approach achieves the best performance and exhibits low training data dependency. Despite the popularity of conversational LLMs, our method outperforms them in this task. Additionally, experiments show that our method maintains good interpretability, a crucial aspect in legal charge prediction tasks.

This study does not explore situations where a single case description may correspond to multiple legal charges. Future research will delve into predictive tasks involving multiple legal charges.

Code availability

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

CRediT authorship contribution statement

Chi Wei: Writing – review & editing, Visualization, Validation, Resources, Data curation. **Jingyun Sun:** Writing – original draft, Software, Resources, Methodology, Formal analysis, Conceptualization.

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.

Acknowledgements

We thank the anonymous reviewers for their in-sightful comments. This work has been supported by the National Natural Science Foundation of China via Grant 62276059, the Heilongjiang Provincial Natural Science Foundation of China via Grant YQ2023F001 and the Special Funds of the National Natural Science Foundation of China via Grant L2424126.

Data availability

Data will be made available on request.

References

[1] H. Wang, T. He, Z. Zou, S. Shen, Y. Li, Using case facts to predict accusation based on deep learning. Proceedings of the 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C), IEEE, 2019, pp. 133–137.
[2] Z. Yang, P. Wang, L. Zhang, L. Shou, W. Xu, A recurrent attention network for judgment prediction. Artificial Neural Networks and Machine Learning–ICANN 2019: Text and Time Series: 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17–19, 2019, Proceedings, Part IV 28, Springer, 2019, pp. 253–266.
[3] H. Chen, D. Cai, W. Dai, Z. Dai, Y. Ding"Charge-based prison term prediction with deep gating network. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 6362–6367..
[4] I. Chalkidis, I. Androutsopoulos, N. Aletras"Neural Legal Judgment Prediction in English. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 4317–4323..
[5] G. Sukanya, J. Priyadarshini, A meta analysis of attention models on legal judgment prediction system, Int. J. Adv. Comput. Sci. Appl. 12 (2) (2021).
[6] L. Gan, Exploiting contrastive learning and numerical evidence for confusing legal judgment prediction. In: Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023, 2023, pp. 12174–12185..

[7] W. Qin, Z. Cao, W. Yu, Z. Si, S. Chen, J. XuExplicitly integrating judgment prediction with legal document retrieval: a law-guided generative approach. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2024, pp. 2210–2220..
[8] Y. Tang, Z. Pan, X. Hu, W. Pedrycz, R. Chen, Knowledge-induced multiple kernel fuzzy clustering, IEEE Trans. Pattern Anal. Mach. Intell. 01 (2023) 1–18.
[9] C. Xiao *et al.*, "Cail2018: A large-scale legal dataset for judgment prediction," arXiv preprint arXiv:1807.02478, 2018.
[10] K.-C. Chien, C.-H. Chang, R.-D. Sun, Legal knowledge management for prosecutors based on judgment prediction and error analysis from indictments, Comput. Law Secur. Rev. 52 (2024) 105902.
[11] H. Zhang, Z. Dou, Y. Zhu, J.-R. Wen, Contrastive learning for legal judgment prediction, ACM Trans. Inf. Syst. 41 (4) (2023) 1–25.
[12] F. Kort, Predicting supreme court decisions mathematically: a quantitative analysis of the 'right to counsel' cases, Am. Political Sci. Rev. 51 (1) (1957) 1–12.
[13] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, Adv. Neural Inf. Process. Syst. 26 (2013).
[14] Z. Hu, X. Li, C. Tu, Z. Liu, M. SunFew-shot charge prediction with discriminative legal attributes. In: Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 487–498..
[15] I. Chalkidis, D. Kampas, Deep learning in law: early adaptation and legal word embeddings trained on large corpora, Artif. Intell. Law 27 (2) (2019) 171–198.
[16] C.-L. Liu, C.-T. Chang, J.-H. Ho, Case instance generation and refinement for case-based criminal summary judgments in Chinese, J. Inf. Sci. Eng. 20 (4) (2004) 783–800.
[17] P. Bhattacharya, K. Ghosh, A. Pal, S. Ghosh, Legal case document similarity: you need both network and text, Inf. Process. Manag. 59 (6) (2022) 103069.
[18] S. Bi, Z. Ali, M. Wang, T. Wu, G. Qi, Learning heterogeneous graph embedding for Chinese legal document similarity, Knowl. Based Syst. 250 (2022) 109046.
[19] L. Gan, K. Kuang, Y. Yang, F. WuJudgment prediction via injecting legal knowledge into neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2021, pp. 12866–12874..
[20] P. Wang, Y. Fan, S. Niu, Z. Yang, Y. Zhang, J. GuoHierarchical matching network for crime classification. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 325–334..
[21] Y. Liu, ML-LJP: multi-law aware legal judgment prediction. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2023, pp. 1023–1034..
[22] Y. Chen, Y. Sun, Z. Yang, H. LinJoint entity and relation extraction for legal documents with legal feature enhancement. In: Proceedings of the 28th International Conference on Computational Linguistics, 2020, pp. 1561–1571..
[23] P. Clark, O. Tafjord, K. RichardsonTransformers as soft reasoners over language. In: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, 2021, pp. 3882–3890..
[24] N. Xu, P. Wang, L. Chen, L. Pan, X. Wang, J. ZhaoDistinguish confusing law articles for legal judgment prediction. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 3086–3095..
[25] C. Xiao, X. Hu, Z. Liu, C. Tu, M. Sun, Lawformer: a pre-trained language model for Chinese legal long documents, AI Open 2 (2021) 79–84.
[26] T. Schick, H. SchützeExploiting cloze-questions for few-shot text classification and natural language inference. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, 2021, pp. 255–269..
[27] H. Wu, B. Ma, W. Liu, T. Chen, D. NieFast and constrained absent keyphrase generation by prompt-based learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2022, pp. 11495–11503..
[28] T. Shin, Y. Razeghi, R.L. Logan IV, E. Wallace, S. SinghAutoPrompt: eliciting knowledge from language models with automatically generated prompts. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 4222–4235..
[29] R. Song, et al., Label prompt for multi-label text classification, Appl. Intell. 53 (8) (2023) 8761–8775.
[30] X. Chen, Relation extraction as open-book examination: retrieval-enhanced prompt tuning. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 2443–2448.
[31] X. Han, W. Zhao, N. Ding, Z. Liu, M. Sun, Ptr: prompt tuning with rules for text classification, AI Open 3 (2022) 182–192.
[32] G. Jiang, S. Liu, Y. Zhao, Y. Sun, M. Zhang, Fake news detection via knowledgeable prompt learning, Inf. Process. Manag. 59 (5) (2022) 103029.
[33] J. Devlin, M.W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of NAACL-HLT, 2019, pp. 4171–4186.
[34] Y. Liu, Roberta: A robustly optimized bert pretraining approach," arXiv preprint arXiv:1907.11692, 2019..
[35] K. Jiang, R. Pradeep, and J. Lin, "Exploring listwise evidence reasoning with t5 for fact verification. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: *Short Papers*), 2021, pp. 402–410.
[36] T. Zhu, Y. Qin, Q. Chen, B. Hu, Y. XiangEnhancing entity representations with prompt learning for biomedical entity linking. In: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, 2021, pp. 4036–4042..
[37] Z. Li, W. Shang, M. Yan, News text classification model based on topic model. Proceedings of the 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), IEEE, 2016, pp. 1–5.
[38] D. Alvarez-Melis, M. SaveskiTopic modeling in twitter: aggregating tweets by conversations. In: Proceedings of the International AAAI Conference on Web and Social Media, 2016, pp. 519–522..
[39] M. Pavlinek, V. Podgorelec, Text classification method based on self-training and LDA topic models, Expert Syst. Appl. 80 (2017) 83–93.
[40] Y. Wang, et al., Topic model based text similarity measure for Chinese judgment document. Data Science: Third International Conference of Pioneering Computer Scientists, Engineers and Educators, ICPCSEE 2017, Changsha, China, September 22–24, 2017, Proceedings, Part II, Springer, 2017, pp. 42–54.
[41] A.J. Rawat, S. Ghildiyal, A.K. Dixit, Topic modeling techniques for document clustering and analysis of judicial judgements, Int. J. Eng. Trends Technol. 70 (11) (2022) 163–169.
[42] R. Silveira, C.G. Fernandes, J.A.M. Neto, V. Furtado, J.E. Pimentel Filho, Topic modelling of legal documents via LEGAL-BERT1, Proceedings 1613 (2021) 0073.
[43] A. Aguiar, R. Silveira, V. Furtado, V. Pinheiro, J.A.M. Neto, Using topic modeling in classification of Brazilian lawsuits. Proceedings of the International Conference on Computational Processing of the Portuguese Language, Springer, 2022, pp. 233–242.
[44] M. Farhadishad, M. Kazemifard, Z. Rezaei, Predicting court judgment in criminal cases by text mining techniques, J. Inf. Technol. Manag. 15 (2) (2023) 204–222.
[45] S.K. Nigam, A. DeroyFact-based court judgment prediction. In: Proceedings of the 15th Annual Meeting of the Forum for Information Retrieval Evaluation, 2023, pp. 78–82..
[46] D. Wang, et al., Knowledge-aware Bayesian deep topic model, Adv. Neural Inf. Process. Syst. 35 (2022) 14331–14344.
[47] L. Yao, et al., Incorporating knowledge graph embeddings into topic modeling, Proc. AAAI Conf. Artif. Intell. (2017).
[48] D. Li, S. Zamani, J. Zhang, P. LiIntegration of knowledge graph embedding into topic modeling with hierarchical dirichlet process. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 940–950..
[49] Y. Zou, A. Pesaranghader, Z. Song, A. Verma, D.L. Buckeridge, Y. Li, Modeling electronic health record data using an end-to-end knowledge-graph-informed topic model, Sci. Rep. 12 (1) (2022) 17868.
[50] H. Zhong, C. Xiao, C. Tu, T. Zhang, Z. Liu, M. SunHow does NLP benefit legal system: a summary of legal artificial intelligence. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 5218–5230..
[51] F. Giunchiglia, M. Yatskevich, P. Shvaiko, Semantic matching: algorithms and implementation. Journal on Data Semantics IX, Springer, 2007, pp. 1–38.
[52] L. Otero-Cerdeira, F.J. Rodríguez-Martínez, A. Gómez-Rodríguez, Ontology matching: a literature review, Expert Syst. Appl. 42 (2) (2015) 949–971.
[53] T. Wang, P. Isola, Understanding contrastive representation learning through alignment and uniformity on the hypersphere. Proceedings of the International Conference on Machine Learning, PMLR, 2020, pp. 9929–9939.
[54] H. Zhong, Z. Guo, C. Tu, C. Xiao, Z. Liu, M. SunLegal judgment prediction via topological learning. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:53083475.
[55] N.E. Zamri, et al., A modified reverse-based analysis logic mining model with weighted random 2 satisfiability logic in discrete hopfield neural network and multi-objective training of modified niched genetic algorithm, Expert Syst. Appl. 240 (2024) 122307.
[56] N.E. Zamri, S.A. Azhar, M.A. Mansor, A. Alway, M.S.M. Kasihmuddin, Weighted random k satisfiability for k= 1, 2 (r2SAT) in discrete hopfield neural network, Appl. Soft Comput. 126 (2022) 109312.