Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Research on legal judgment prediction model based on event extraction and constraints

Shan Cong[1,] ✉

[1] *School of Law, Dongguan City University, Dongguan, Guangdong, 523000, China*

ABSTRACT

Research on event extraction and constraint encoding of legal cases, using Lawformer as a pre-trained language model for legal sentence prediction model, constructing MJP-Law model to predict the sentence of legal cases. The HAN encoder in the model is utilized to extract the inter-sentence relations in the legal case and construct the relations among the law, the charge, and the sentence period. Compare the performance of this paper's MJP-Law model with other prediction models on law, charge, and sentence period, and explore the effects of the three subtasks of law, charge, and sentence period on the model through ablation experiments, and compare the prediction effects of a single MJP model and the MJP-Law model on low-frequency charges. In this paper, the MJP-Law model outperforms other prediction models in terms of prediction performance on statute, offense, and sentence. The four models of "MJP-Law", "MJP-Law_law", "MJP-Law_SG" and "MJP" had the same prediction performance, which were 95.54%, 89.86%, 89.73% and 89.81%, respectively. "MJP-Law" and "MJP-Law_law", "MJP-Law_SG" and "MJP" have the same performance in law prediction. After removing the sentencing guidelines and legal sentences, the macro F1 values of the MJP-Law model all showed a decrease.The predictive performance of the MJP-Law model on low-frequency offenses was better than that of the single MJP model.

Keywords: event extraction, constraint coding, MJP-Law model, legal sentence

1. Introduction

Legal sentence prediction is an important task in the field of legal artificial intelligence, which aims to predict the sentence results of the legal provisions, charges, and sentences involved in the defendant through factual descriptions. Artificial intelligence has now become an important force to promote technological innovation and industrial change, and its intelligent and automated characteristics have powerfully improved the efficiency of all walks of life [1-4]. The legal field is no exception, with the continuous development of society and the improvement of citizens' legal awareness, the number of

✉ Corresponding author.
E-mail address: lisacong43@outlook.com (S. Cong).
Received 12 February 2024; Revised 15 April 2024; Accepted 25 November 2024; Published Online 15 April 2025.
DOI: 10.61091/jcmcc127a-176

cases accepted by courts at all levels has increased year by year, and the contradiction of "too many cases, too few people" has brought a heavy workload to law-related practitioners. In order to alleviate their work pressure, intelligent justice came into being [5-8].

Legal judgment prediction is an important task in intelligent justice, and its main purpose is to predict the possible verdict of the defendant based on the factual description of the legal documents, including the applicable legal provisions, suspected crimes, and the possible sentence and other information, which has been widely concerned by artificial intelligence researchers and legal practitioners for decades [9-12]. The legal judgment prediction technology alleviates the problems of heavy burden and pressure of law-related practitioners to a certain extent, and is a key technology of the intelligent legal assistant system. On the one hand, it provides convenient references for lawyers, judges and other professionals and improves work efficiency [13-15]. On the other hand, it can avoid the judgment bias caused by the non-uniformity of sentencing scale between different judges, and effectively solve the problems of different judgments in the same case and similar cases not similarly handled [16-17].

Literature [18] emphasizes the important role of legal judgment prediction for judges, lawyers, and parties. The legal judgment prediction algorithms released from 2018 to 2022 were investigated, especially the deep learning, evaluation measures, limitations and other factors involved. And based on the classification method, the collected information was classified into two channels, criminal and civil cases, and analyzed the class again according to the prediction techniques. Literature [19] states that the purpose of legal judgment prediction is to automatically predict the judge's decision under different legal conditions. Applying the literature review method to summarize the relevant studies aims to provide an overview of the relevant studies that combine numerous elements of legal judgment prediction, including datasets, methodologies, and so on. Literature [20] describes that the purpose of legal judgment prediction is to identify information based on factual findings in order to determine the outcome of a judgment. The THME network was proposed to address the shortcomings of legal judgment prediction, which was used in a large-scale dataset of criminal cases, and the results showed that the method was superior to legal judgment prediction in judgment prediction work. Literature [21] designed a legal judgment prediction method combined with matching guiding cases after collecting relevant typical cases, and proposed a multi-feature fusion legal judgment prediction model, MAGiC, aiming to improve the performance of legal judgment prediction. The experimental results verify that the method effectively improves the performance of the legal judgment prediction model. Literature [22] mentions that the development of AI legal judgment prediction model will help the court proceedings due to the increasing number of cases one by one. And the integration of the model with the judicial process will not only improve the efficiency of the court, but also legal practitioners will benefit from this model.

Literature [23] uses machine learning models to predict the IPC portion of a case that may be applicable by analyzing the basic description of the case, a process that can provide messages such as penalties and charges. The results of the legal verdict prediction help non-legal people to better understand the case, in addition to facilitating decision-making by judges and lawyers. Literature [24] mentions that automated legal judgment prediction has received a lot of attention in the field of natural language processing in recent years, partly due to the importance of having a good practical value. The results of research on legal judgment prediction in multiple jurisdictions and multiple languages are summarized and their research directions are discussed. Literature [25] introduces a rationale-based legal judgment prediction framework based on the real judgment logic of judges, and demonstrates through extensive experiments on real datasets that the framework performs better than other methods and has more practical value due to its good interactivity and interpretability. Literature [26] discusses the application of machine learning algorithms in the legal litigation process. It is indicated that the CNN+LSTM model is more helpful for the judiciary to handle the task as compared to the manual one, which has a higher accuracy rate as compared to the manual one.

Recommendations were made for the effective utilization of AI techniques in judicial cases. Literature [27] proposed a novel method for predicting civil case decisions by simulating the logic of the judge, constructed an external knowledge base including the explanation and application of each reason, and designed an encoder layer and an interaction layer, which was verified to be effective through experiments. Literature [28] indicates that correctly predicting court decisions is a complex process. The application of machine learning algorithms such as plain Bayes and logistic regression in order to predict legal cases was utilized and the accuracy of the classifier and the confusion matrix were evaluated. Literature [29] proposes a new legal judgment prediction framework which explicitly incorporates retrieved historical cases for legal judgment prediction. It is proved through experiments on commonly used legal datasets that the model outperforms other baseline models.

The article first combs through the relevant elements of the event extraction concept and selects the evaluation metrics. The pre-trained language model is used as the encoder of the prediction model to constrain the decoding of the event extraction results. Then Lawformer is used as the pre-trained language model of the legal sentence prediction model in this article, and the HAN model is utilized to capture the inter-sentence relationship of the legal case and encode the relevant crime composition. The relationships among the three subtasks of law, crime, and sentence are constructed to simulate the judge's trial of the case to construct the Lawformer-based legal sentence prediction model (MJP-Law). Compare the prediction results on the law, charge, and sentence of the MJP-Law model in this paper with other prediction models to verify the prediction performance of the MJP-Law model. Explore the effect of relevant modules in the model through ablation experiments and analyze the predictive effect of the model on low-frequency charges.

2. Event extraction and constraints

2.1. Event Extraction Tasks

2.1.1. Relevant concepts. As natural language processing techniques continue to advance, event extraction techniques are increasingly being used and studied [30].

The following are definitions of some key terms used in event extraction tasks:

Event: refers to a specific event narrated in a piece of text, which usually includes the event's trigger word (a sign of the event's occurrence) and the event's thesis element (the key information of the event).

Trigger word: refers to a special word or phrase that is a grammatical or semantic marker for some kind of event (e.g., arrest, explosion, occurrence). Trigger words are used to identify the location of a possible event in the text and to extract further details about the event (e.g., participants in the event, location, time, etc.).

Event Thesis Elements: refers to the components of an event, including information about the subject, time, location, etc. of the event. This information is usually organized into a fixed template for describing various aspects of the event. For example, an event thesis element may describe “the subject of something that happened at a certain time in a certain place”. By annotating events in text with theses, event information can be better understood and analyzed.

Event extraction refers to the extraction of event information from unstructured text, such as people, places, time and other event elements, and the identification and classification of event types. According to the different objectives of the task and the different datasets, the event extraction task can be divided into different subtasks:

Event Trigger Word Recognition: identifying words or phrases in the text that may indicate the occurrence of an event, mostly verbs or nouns.

Event Trigger Word Classification: classifying the events indicated by event trigger words into different types, it is a typical classification problem, which focuses on the detection of event sentences and the classification of event sentences.

Event Argument Recognition: refers to recognizing event-related arguments from natural language texts, including arguments of trigger words and arguments of other semantic roles (e.g., time, place, participants, etc.).

Event Argument Classification: the goal of event argument classification is to categorize the identified event arguments to determine the type of semantic roles they belong to. For example, an event argument can be classified as "person", "organization", "place", etc. The event triggers are identified and the event triggers are classified into the following categories.

Event trigger word recognition and event trigger word classification can also be referred to as event recognition or event detection tasks.

2.1.2. Evaluation of indicators. If the position of the prediction made for the trigger word agrees with the position of the labeling made, the trigger word identification made is judged to be correct. After the trigger word is accurately identified, if the event subtype corresponding to the predicted trigger word agrees with the result of manual labeling, then it is judged as the correct trigger word classification.

In event extraction, three items, precision (P), recall (R) and F1 value (F1), are usually used as the basic evaluation metrics. Among them, precision in which refers to the ratio of the number of correct instances extracted by the model to the total number of extracted instances, this metric is used to measure the accuracy of the event extraction model. Recall refers to the ratio of the number of correct instances extracted by the model to the total number of correct instances, which is used to measure the comprehensiveness of the event extraction model, and F1 value refers to the weighted average of precision and recall, which is used to measure the overall performance of the event extraction model:

$$Precession = \frac{TP}{TP + FP} \times 100\% \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \tag{2}$$

$$F1 = \frac{2TP}{2TP + FP + FN} \times 100\% \tag{3}$$

Where TP (true cases) refers to the number of correctly extracted and predicted positive cases, FP (false positive cases) refers to the number of incorrectly extracted but predicted positive cases, and FN (false negative cases) refers to the number of incorrectly extracted but predicted negative cases. The magnitude of F1 is generally used to measure the superiority of event extraction methods. In event extraction, the evaluation metrics do not only consider the correctness of the event type, but also the completeness and correctness of the event thesis elements. Therefore, the evaluation needs to evaluate the information of event type, event thesis element, and trigger word.

2.2. Pre-training language models

The event extraction task is an NLP task, so it requires advanced natural language processing techniques in addition to an understanding of events and event-related knowledge. The abstract mathematical model established for the objective characteristics of natural language is called a language model (LM), which is an abstract tool for describing natural language and is a key technology to make natural language understandable and analyzable by computers. Natural language is a symbol-based information medium created by human beings, and unlike digital signals such as images and audio, symbols in natural language are discrete and do not have spatial continuity. This also leads to the fact that in the early stage of modeling for natural language can only be done by studying the complex grammar syntax laws, which results in the construction of natural language processing system is costly, and the information is redundant, and the flexibility is very poor.

After the birth of word embedding technology, natural language has been able to take advantage of

deep learning, such as convolutional neural networks, Long Short-Term Memory (LSTM) networks with stronger sequence encoding ability, and Gated Recurrent Unit (GRU) neural networks, which have greatly improved the performance of various NLP tasks. However, word representations obtained by word embedding techniques are static and have no way to cope with the case of multiple meanings of a word, and inspired by the field of computer vision, the idea of constructing a pre-trained language model that can dynamically adapt to contextual environments to replace static word embeddings has begun to emerge.The ELMo bi-directional language model utilizes a bi-directional LSTM as an encoder of the text on top of pre-trained word embeddings, and outputs the word vector representations with contextual context-aware word vector representation.

However, since the encoder used in ELMo is LSTM, its context-awareness and training time are affected by the recursive algorithm, and LSTM is not a suitable encoder for pre-training language models. In the same period, Transformer, which is based on multi-head attention mechanism for text encoding, greatly improves the above problems, and breaks the distance limitation of information transfer between contexts through the attention mechanism for context-awareness and realizes the parallelism of text encoding. So the language model based on Transformer began to appear.

Thanks to the powerful ability of Transformer, and the increase of training corpus, the subsequent pre-training language models with excellent performance basically take Transformer as the basic model, and transfer more energy to the setting of pre-training tasks, the setting of language model and the construction of corpus. Due to its large parameter scale and large-scale corpus participation in the pre-training phase, the pre-trained language model has the ability to far exceed the expectations of the initial distributed feature representation. At present, the pre-trained language model has the potential capability and foundation to solve various NLP tasks, and the research after this paper will also be carried out with the Transformer-derived model as the basic encoder.

2.3. Constraint Decoding

Constraint decoding is the process of placing restrictions on the generated results when generating event information in a generative event extraction task [31]. Typically, generative models generate a large amount of event information, many of which do not conform to predetermined rules, are illegal or unreasonable, so constraint decoding is needed to ensure that the generated event information has a certain degree of legitimacy and reasonableness. The method of constraint decoding can be to filter the generated events or to introduce certain restriction rules in the generation process. Constraint decoding can improve the accuracy and reasonableness of the generated event information, and it can also help the model learn the correct generation strategy. Through constraint decoding, the generated event information has higher correctness and better meets the actual demand, it can also limit the generation space of event information, which makes the generation of the model more efficient, it can standardize the generation format of the event information and improve the readability of the generated event information.

3. Lawformer-based legal judgment prediction model construction

This chapter proposes a new multi-task judgment prediction model MJP-Law based on Lawformer.The structure of the MJP-Law model is shown in Fig. 1, which includes a case description coding module, a relevant crime composition coding module, and a multi-task learning module.

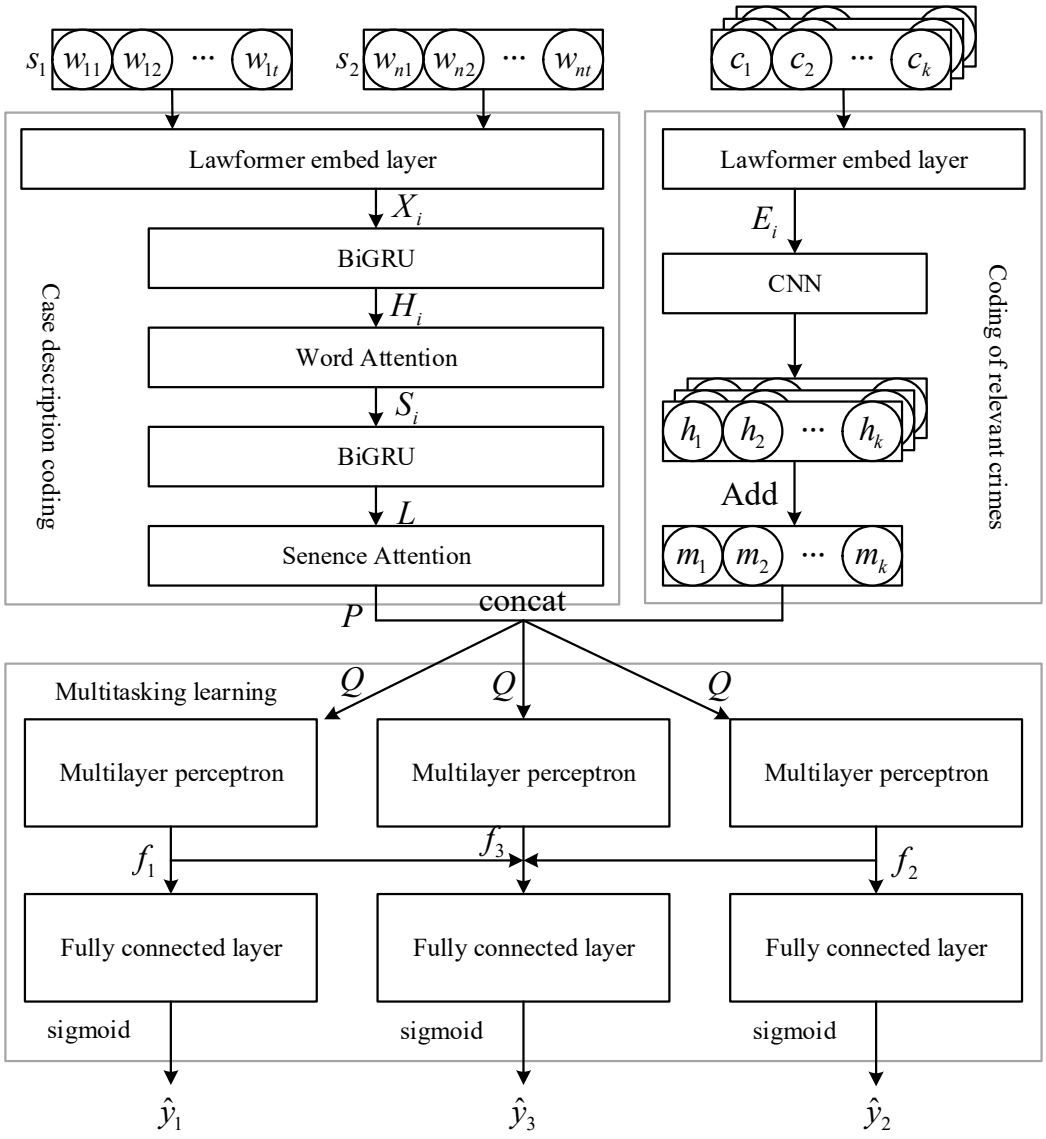

Fig. 1. MJP-Law model structure

3.1. Lawformer word embeddings

Lawformer is the first pre-trained model designed specifically for the judicial domain [32]. Three attentional mechanisms are included in Lawformer, namely sliding window attentional mechanism, extended sliding window attentional mechanism and global attentional mechanism.

Given the case description text $d = \{s_1, s_2, ..., s_n\}$, n is the number of sentences, the i rd sentence $s_i = \{w_{i1}, w_{i2}, ..., w_{ii}\}$, w_{ij} denotes the j th word in the i th sentence, and t denotes the sentence length. w_{ij} After Lawformer word embedding, we get the vector representation x_{ij}, which is calculated as:

$$x_{ij} = Lawformer(w_{ij}) \tag{4}$$

3.2. HAN Encoder

The case description text belongs to long text, which contains multiple types of information elements such as the cause, passage, and result of the case with typical hierarchical structure, and the HAN model is utilized to capture the inter-sentence relationship of the text.The HAN encoder consists of word level coding module and sentence level coding module.

3.2.1. Word-level coding. The word level coding module uses a model structure that combines BiGRU and Attention to obtain a vector representation of a sentence by coding a sequence of word levels.

The BiGRU model is able to encode from both forward and reverse directions to learn contextual information. Sentence $X_i = \{x_{i1}, x_{i2}, ..., x_{ii}\}$ after word embedding is taken as the input of BiGRU, which is encoded in both directions to obtain the hidden layer state $\vec{h}_{ij}, \overleftarrow{h}_{ij}$ respectively, and is spliced to obtain the output h_{ij}. The computational procedure is as follows:

$$\overrightarrow{h_{ij}} = \overrightarrow{GRU}(x_{ij}) \tag{5}$$

$$\overleftarrow{h_{ij}} = \overleftarrow{GRU}(x_{ij}) \tag{6}$$

$$h_{ij} = [\vec{h}_{ij}, \overleftarrow{h}_{ij}] \tag{7}$$

In order to find the words that contribute more to the semantic expression of the sentence, the attention mechanism is introduced for encoding. The attention mechanism assigns weights to each word in the sentence according to the degree of contribution. Real words have a higher impact on the semantic expression of the sentence and will be assigned higher weights; these words represent the core semantics of the sentence. Imaginary words have less impact on the semantic expression of the sentence and will be assigned lower weights and do not affect the core semantic expression of the sentence.

Specifically, the hidden representation u_{ij} is first obtained by a single-layer perceptron, and then a normalized exponential function (Softmax function) is used to calculate the association similarity between the hidden representation u_{ij} and the vector representation u_w of the sentence context, which in turn obtains the corresponding probabilistic weights α_{ij}, α_{ij} representing the degree of contribution of the word in the corresponding position to the semantic expression of the sentence, and u_w representing the core semantics of the sentence, from the random initialization of the model parameters that are obtained by the The update is realized in the neural network training. Finally, the vector weighted summation is used to obtain the final vector representation of the sentence S_i. The specific computational procedure is as follows:

$$u_{ij} = \tanh(W_w h_{ij} + b_w) \tag{8}$$

$$\alpha_{ij} = \frac{\exp(u_{ij}{}^T u_w)}{\sum_j \exp(u_{ij}{}^T u_w)} \tag{9}$$

$$S_i = \sum_j \alpha_{ij} h_{ij} \tag{10}$$

where W_w and b_w are the weight matrix and bias vector, respectively.

3.2.2. Sentence-level coding. Similarly, the sentence level encoding module adopts the model structure combining BiGRU and Attention, where sentences form a text paragraph, and the feature extraction of the text paragraph information is accomplished by encoding the sentence vector S_i.

Firstly, the vector representation s_i of the sentence is input into BiGRU, and after two-way encoding, the hidden layer state $\vec{L}_i, \overleftarrow{L}_i$ is obtained respectively, and the two vectors are spliced to get the output L_i. The computational process is shown as follows:

$$\overrightarrow{L_i} = \overrightarrow{GRU}(S_i) \tag{11}$$

$$\overleftarrow{L_i} = \overleftarrow{GRU}(S_i) \tag{12}$$

$$L_i = [\vec{L}_i, \overleftarrow{L}_i] \tag{13}$$

Attention mechanism is used to measure the importance of each sentence in the text, assigning higher weights to sentences with high importance and lower weights to sentences with low importance.

Firstly, the hidden representation u_i is obtained by single-layer perceptron, and then the Softmax probability function is used to calculate the association similarity between the hidden representation u_i and the text passage representation u_s to obtain the weight of the corresponding sentence α_i, u_s represents the core semantics of the text passage, which is obtained by random initialization, and parameter updating is carried out by backpropagation. Finally, the vector representation h_i of each sentence and the corresponding weight α_i are weighted and summed to get the text paragraph representation P. The computational procedure is shown below:

$$u_i = \tanh(W_s L_i + b_s) \tag{14}$$

$$a_i = \frac{\exp(u_i^T u_s)}{\sum_i \exp(u_i^T u_s)} \tag{15}$$

$$P = \sum_i a_i h_i \tag{16}$$

where b_s and W_s denote the bias vector and weight matrix of the single-layer perceptron, respectively.

3.3. Relevant Criminal Composition Codes

Crawl the corresponding criminal composition of the charges on ChinaFindLaw.com, go to classify the charges by binary classification, input the case description into the support vector machine classifier, according to the output of the probability magnitude of each charge, get the m charge that has relevance to the case, and find the corresponding criminal composition through these charges. The relevant crime composition C_i is embedded to get the word embedding representation $E_i = \{e_1, e_2, \ldots, e_k\}$. where, $i \in [1, m]$, k are the number of words. It is expressed by the following formula:

$$E_i = Lawformer(C_i) \tag{17}$$

CNN is used to learn the textual features that constitute the relevant crime. The computational formula is shown below:

$$h_i = W_m \otimes e_{j:j+h-1} + b_m, j \in [1, k-h+1] \tag{18}$$

Where "$\otimes$" is the convolution operation, W_m and b_m denote the convolution matrix and bias vector respectively, and h is the size of the upper and lower sliding windows.

Since these m crime components are related to the same case description, in order to capture the associated features with the case description, vector summation is used to obtain the related crime component coding representation of the case description M. The computational formula is shown below:

$$M = \sum_m h_i \tag{19}$$

The vector representation P obtained at the case description coding end is further vector spliced with the vector representation M obtained at the crime composition coding end. This is represented by the following equation:

$$Q = [P, M] \tag{20}$$

3.4. Multi-task learning module

In order to simulate the judge's trial thinking, multiple fully connected layers are used to construct the relationship between the three subtasks of law, charge and sentence.

In predicting the charge, the splicing vector Q is used to obtain f_1 through a multilayer perceptron, and f_1 is used as a specific representation of the charge prediction task, using the fully connected layers as classifiers, and the output is transformed into the prediction probability of each category of the charge using a sigmoid activation function. The formula is calculated as follows:

$$\hat{y}_1 = \text{sigmoid}(W_1 f_1 + b_1) \tag{21}$$

where b_1 and W_1 denote the bias vector and weight matrix of the classifier for the offense prediction task, respectively, and $\hat{y}_1$ denotes the prediction probability of each category of the offense.

Similarly, in predicting the statute, the splice vector Q is passed through a multilayer perceptron to obtain f_2, f_2 is used as a specific representation of the statute prediction task, a fully connected layer is used as a classifier, and the output is transformed into the predicted probability of each category of the statute using a sigmoid activation function. The computational formula is shown:

$$\hat{y}_2 = sigmoid(W_2 f_2 + b_2) \tag{22}$$

where b_2 and W_2 denote the bias vector and weight matrix of the classifier for the French bar prediction task, respectively. $\hat{y}_2$ denotes the prediction probability of each category of the statute.

In predicting the sentence, the splicing vector Q is passed through the multilayer perceptual machine to obtain f_3, and f_1, f_2 and f_3 are made to be spliced, and the splicing is used as a specific representation of the sentence prediction task, using the fully connected layer as a classifier, and the output is transformed into the prediction probability of each category of the sentence using the sigmoid activation function. The formula is as follows:

$$\hat{y}_3 = \text{sigmoid}\left(W_3 \cdot (f_1, f_2, f_3) + b_3\right) \tag{23}$$

where W_3 and b_3 are the weight matrix and bias vector of the classifier for the sentence prediction task, respectively, and $\hat{y}_3$ denotes the prediction probability of each category of the sentence.

For each sub-task, here the loss is calculated separately using the binary cross entropy loss function, assigning weights to the loss of each task, and finally the total loss is calculated using weighted summation. The calculation formula is as follows:

$$Loss = -\sum_{i=1}^{3} \lambda_i \sum_{j=1}^{|Y_i|} (y_{i,j} \log_2(\hat{y}_{i,j}) + (1 - y_{i,j}) \log_2(1 - \hat{y}_{i,j}) \tag{24}$$

where $y_{i,j}$ and $\hat{y}_{i,j}$ are the probabilities of one-hot coding and predicted labels corresponding to the real labels of class j of the i rd task, respectively, and $|Y_i|, \lambda_i$ denotes the total number of labels and the weight parameter of the i th task, respectively.

4. Analysis of the results of the experiment

4.1. Experimental setup

In this section, extensive experiments are conducted on two real-world datasets in order to investigate the effectiveness of the model proposed in this chapter in the judgment prediction task. First, this section describes the datasets and the preprocessing of the data. Then, some necessary parameters are provided to better understand the model, followed by comparing the experimental results with other prediction models. Finally, ablation experiments are conducted to analyze the impact of each module in the model.

4.1.1. Data set description. The data used for the experiments in this section comes from CAIL2018 first published in the 2018 China Artificial Intelligence and Law Challenge Cup competition, which collects more than 5.7 million criminal documents issued by China's Supreme People's Court constitute. After some cleaning, these data were randomly divided into two complete datasets CAIL-Small, CAIL-Big. Where this dataset is stored in the format of json, which includes the security description, the legal text used, the charge and the sentence length.

4.1.2. Experimental setup and evaluation indicators. In this section, the experimental setup will be described and the evaluation metrics during the training process will be presented.When the MJP-Law model was trained, the authors of this paper used the Adam optimizer, where the learning rate was set to 10-4, the corresponding weight decay to 10-6, and the dropout probability was set to 0.5.The authors of this paper performed the model for 30 epochs training, while setting the batch size of 128 and using word2vec for word embedding dimension of 200. at the same time, the maximum length of the text for factual description, legal text semantics and charge definition information is set to 350, 150 and 300 respectively, which can cover 80% of the samples. For the inference layer, the top-k with a k-value of 5 indicating the largest probability value of sentence prediction is selected as the top 5 candidate items. Finally, accuracy (Acc), macro precision (MP), macro recall (MR) and macro F1 (MF1) are used to evaluate the final model.

4.1.3. Methods of comparison. To evaluate the performance of the proposed method, MJP-Law model is compared with the following methods:
1) SVM: This technique uses word2vec word embedding followed by SVM as a text classifier.
2) FLA: Luo et al. use civil law countries can extract the legal text can provide a legal basis for the prediction of the crime, so a unified model is proposed to jointly model the crime, legal text and sentence prediction task using neural network methods to capture the interaction between factual descriptions and knowledge of the applicable law and combined to complete the prediction of legal sentences.
3) TOPJUDGE: Zhong et al. simulate the case logic of a human judge and model the subtasks as topological dependencies and further propose a unified multitask learning model that incorporates topological dependencies into a unified model, which greatly affects the credibility and interpretability of trial predictions.
4) LADAN: Xu et al. found that the work that exists can well extract semantic information from case descriptions, but still face the problem of confusion between the charges and legal texts, so propose a model based on attention and similarity graphs to use the differences between the charges (legal texts) to differentiate between the confusing charges (legal texts) to achieve improved modeling results.
5) NEURJUDGE: Yue et al. use the intermediate results of the subtask of legal judgment prediction to divide the factual descriptions into different cases, and use the method of label embedding to combine the semantics of the information of the charge and the legal provisions into the case facts, to generate a more expressive factual representation to predict the subtask.

4.2. Performance analysis

The results of the prediction of the different models on the CAIL dataset for the legal texts, charges and sentences are shown in Tables 1 to 3, respectively. From the analysis of Tables 1, 2, and 3, MJP-Law achieves the best results, which is not only better than FLA, but also better than TOPJUDGE, LADAN, and NEURJUDGE.The MJP-Law model performs the best among all the legal sentence prediction models, which may be attributed to the fact that MJP-Law can simulate the process of a real courtroom sentence well, and relying on multiple subtasks dependencies and constraint relationships can well improve the performance of the model. Table 3 shows that there is also an improvement in sentence prediction, which may be due to the fact that the severity of the circumstances provides favorable

assistance in sentence prediction.All the methods in the CAIL-Big dataset outperform CAIL-Small, which is due to the fact that the training data of CAIL-Big is more sufficient.The MJP-Law model has a more significant improvement in the sub-tasks of charge prediction and sentence prediction, which is mainly because the consistency constraint of the task labels is used to predict the sentence and charge. Is the use of consistency constraints on task labels to improve accuracy, and for sentence prediction utilizes logical reasoning on the severity of the plot to improve the robustness of the model.

Table 1. Results of method legal article prediction for different models on the CAIL-Dataset (%)

Model	Legal article							
	CAIL-Small				CAIL-Big			
	Acc	MP	MR	MF1	Acc	MP	MR	MF1
SVM	80.63	79.91	77.57	77.79	95.25	80.81	78.54	78.84
FLA	83.40	80.40	76.75	80.28	94.82	80.72	77.97	80.44
TOPJUDGE	80.45	77.00	78.91	79.83	91.01	88.28	79.44	79.96
LADAN	83.61	77.17	78.29	79.54	93.28	77.23	80.76	80.27
NEURJUDGE	79.74	83.25	76.43	80.65	94.49	84.25	77.19	81.37
MJP-Law	84.47	85.59	81.41	82.32	96.77	89.68	88.67	89.09

Table 2. Results of method charge prediction for different models on the CAIL-Dataset (%)

Model	Charge							
	CAIL-Small				CAIL-Big			
	Acc	MP	MR	MF1	Acc	MP	MR	MF1
SVM	81.43	76.63	78.50	81.98	87.64	77.16	79.32	82.21
FLA	80.50	77.36	78.96	87.39	85.65	80.34	79.60	88.76
TOPJUDGE	84.56	81.53	83.10	86.92	94.02	82.88	83.41	87.42
LADAN	85.85	79.32	82.88	79.80	88.92	87.92	83.84	81.71
NEURJUDGE	81.70	78.15	79.26	82.49	88.77	84.75	79.93	83.47
MJP-Law	89.79	85.38	83.70	88.12	95.54	89.86	89.73	89.81

Table 3. Results of method term of penalty for different models on the CAIL-Dataset (%)

Model	Term of penalty							
	CAIL-Small				CAIL-Big			
	Acc	MP	MR	MF1	Acc	MP	MR	MF1
SVM	34.47	26.02	26.56	28.34	40.57	39.44	38.89	40.29
FLA	37.13	31.88	29.50	30.18	44.30	40.39	40.29	43.65
TOPJUDGE	37.94	29.82	40.36	38.49	50.83	42.06	42.27	44.72
LADAN	39.66	29.87	37.24	39.73	47.91	44.22	41.39	41.45
NEURJUDGE	40.21	30.98	38.27	35.81	50.10	46.60	40.81	45.45
MJP-Law	42.27	35.70	40.89	41.87	55.53	48.28	46.40	47.02

4.3. Analysis of ablation experiments

4.3.1. Characterization de-correlation module effects. In order to verify the effect of the module combining the sentencing guidelines, and the LJP module combining the legal provisions on the model performance respectively, this paper conducts ablation experiments, and the results are shown in Table 4. Where “MJP-Law” is the performance of the current model on the three subtasks, “MJP-Law_law” denotes the result of removing the module that combines legal provisions, “MJP-Law_SG” denotes the result of removing the sentencing guideline module. From the results of the ablation experiments, it can be seen that the removal of each module has a certain impact on the model performance, indicating that these two modules play an important role in the current model “MJP-Law”.

For the statute prediction task, a multi-task learning approach was initially used, consisting of three subtasks, statute prediction, offense prediction, and sentence prediction, which were able to share weights and feature representation space during training. However, in the current experiment the statute prediction task was separated separately and prioritized before proceeding to the remaining two tasks. According to the results, the performance of "MJP-Law" and "MJP-Law_law" and "MJP-Law_SG" and "MJP" is the same in terms of legal prediction. The prediction accuracy of "MJP-Law" and "MJP-Law_law" were 96.77%, 89.68%, 88.67% and 89.09%, respectively, and the prediction results of "MJP-Law_SG" and "MJP" were 96.26%, 89.15%, 88.16% and 88.58%, respectively. This suggests that the performance gain from splitting the statute prediction task is not due to the fact that the initial three tasks were trained together in a design where statute prediction, charge prediction, and sentence prediction shared features effectively to some extent, and some of these features assisted in the learning process of the statute prediction task. Therefore, when the law prediction task is split separately, although it gains a separate feature representation, it also loses the auxiliary learning effect of the other tasks, and this loss offsets the potential performance improvement brought by the separate feature representation, resulting in no change in the overall performance. Comparing “MJP-Law” with “MJP-Law_SG”, it can be found that the macro F1 value of the task decreased by 0.51% when the sentencing guideline was removed, which indicates that the sentencing guideline has a certain effect on the task of predicting the law.

For the crime prediction task, removing any of the modules did not affect the performance of the model, which may be due to the fact that the model has already fully learned and understood the feature information needed for crime prediction in the original task, and does not need to be further improved by the assistance of new modules. Second, the performance of the crime prediction task itself is already very high, so the addition of new modules does not bring significant improvement.

In the sentence prediction task, comparing “MJP-Law” and “MJP-Law_law”, we can find that the macro F1 value decreases by 0.59% after removing the legal text and replacing it with the method of three subtasks co-training. This indicates that the benchmark-type relevant legal provisions defined in the legal text play a positive role in the sentence prediction task. When the Sentencing Guidelines

module was removed, all performance metrics decreased significantly, including a decrease of 0.93% in the macro F1 value, which shows that the Sentencing Guidelines module plays a crucial role in the sentence prediction task. The module contains guidance details for various sentencing scenarios, which can provide richer background information and finer judgment basis for the complex sentence prediction problem. Taken together, compared to the single MJP model, the introduction of the two modules, legal provisions and sentencing guidelines, significantly improves the performance of the model, with a 1.29% increase in the macro F1 value, which can be attributed to the legal provisions provided by the legal provisions as well as the specific guidance for a variety of sentencing scenarios in the sentencing guidelines module, which is crucial for the model to understand and deal with the complex sentencing prediction problem.

Table 4. Ablation experiment results (%)

		MJP-Law	MJP-Law_law	MJP-Law_SG	MJP
Legal article	Acc	96.77	96.77	96.26	96.26
	MP	89.68	89.68	89.15	89.15
	MR	88.67	88.67	88.16	88.16
	F1	89.09	89.09	88.58	88.58
Charge prediction	Acc	95.54	95.54	95.54	95.54
	MP	89.86	89.86	89.86	89.86
	MR	89.73	89.73	89.73	89.73
	F1	89.81	89.81	89.81	89.81
Term of penalty	Acc	55.53	55.02	52.11	51.20
	MP	48.28	47.86	47.28	46.87
	MR	46.40	46.08	45.82	45.13
	F1	47.02	46.43	46.09	45.73

4.3.2. Analysis of low-frequency charges. In this paper, we visualize the confusion matrices of the prediction results of some low-frequency charges.The results of the confusion matrix visualization of the prediction results of the MJP and MJP-Law models are shown in Figures 2 and 3, with the vertical axis denoting the true labels of the models and the horizontal axis denoting the prediction results.

As mentioned in the previous section, data imbalance is a great challenge for crime prediction, in order to see more intuitively the performance of the model proposed in this paper on low-frequency crimes, this paper first determines the low-frequency crimes by analyzing the prediction results and the true labels, and selects the data of the low-frequency crimes from the test set for testing. The confusion matrix of the prediction results using the MJP model is shown in Figure 2, and Figure 3 represents the confusion matrix of the prediction results using the MJP-Law model proposed in this paper. It can be observed that the prediction effect of MJP-Law on low-frequency offenses is higher than the prediction effect of the single MJP model. The prediction accuracy of MJP-Law reaches 100% in eight low-frequency crimes except “other”, while the MJP model only reaches 100% in the prediction of “perjury”, and MJP-Law shows better prediction ability than the MJP model, which strongly verifies the validity of the proposed method. MJP-Law shows better prediction ability than MJP model, which strongly validates the effectiveness of the method proposed in this paper.

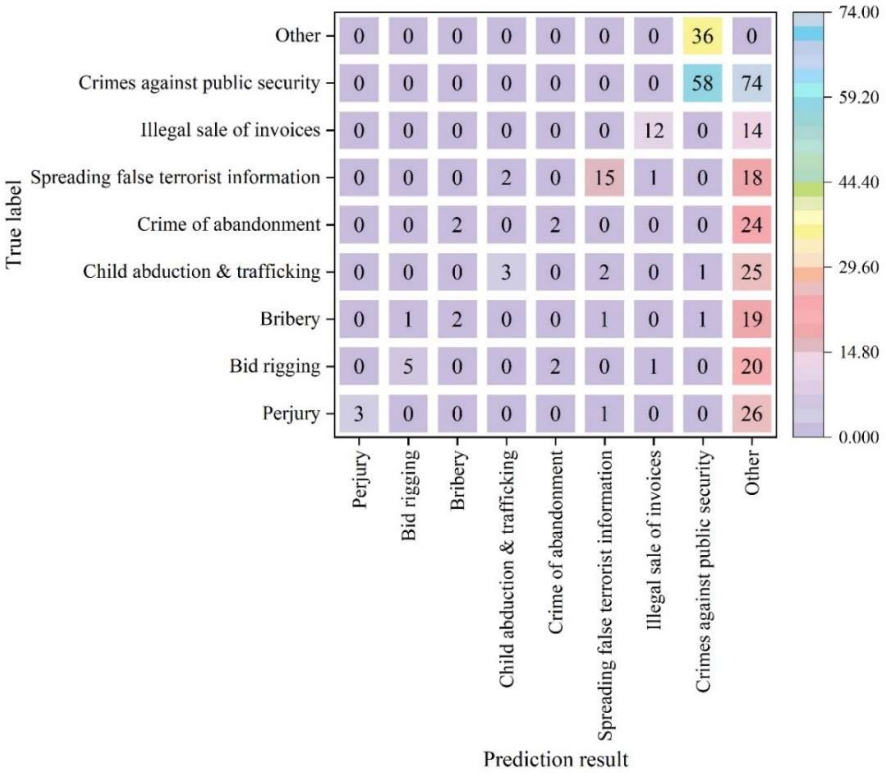

Fig. 2. MJP model prediction result confusion matrix

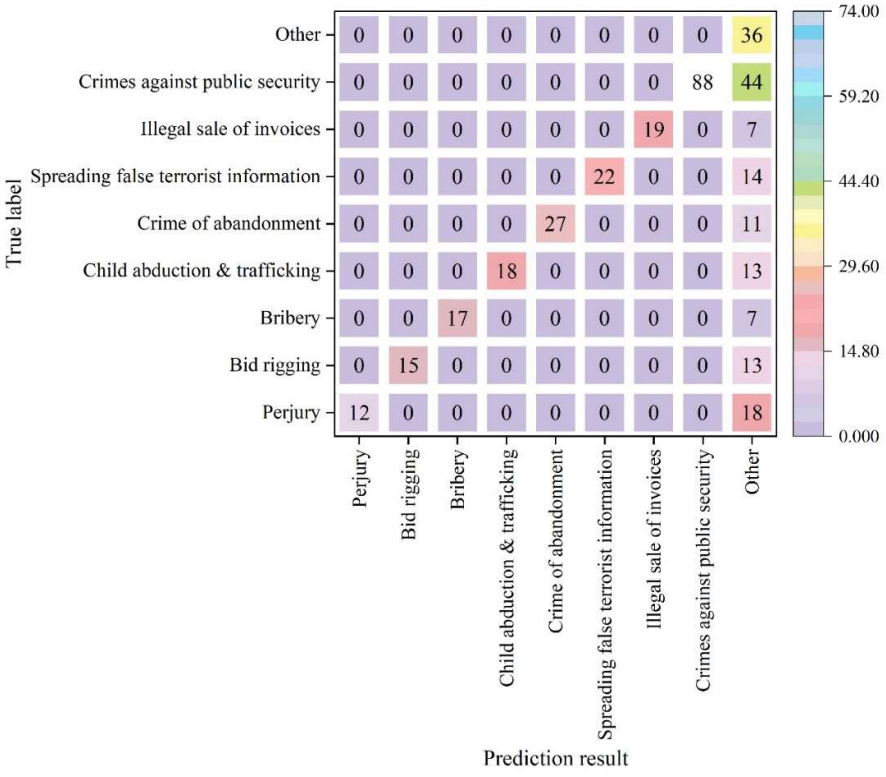

Fig. 3. MJP-Law model prediction result confusion matrix

5. Conclusion

In this paper, we construct Lawformer-based legal sentence prediction model (MJP-Law) based on

event extraction and constraints, design experiments to compare the performance of MJP-Law model with other legal sentence prediction models, and carry out ablation experiments to verify the effectiveness of each module.

The MF1 values of this paper's MJP-Law model in CAIL-Big dataset for the prediction of statute, offense and sentence are 89.09%, 89.81% and 47.02%, respectively, which are higher than other prediction models. The prediction accuracy of the MJP-Law model in this paper achieved the best results among all models.

"MJP-Law", "MJP-Law_law", "MJP-Law_SG", and "MJP" had the same performance in crime prediction, which were 95.54%, 89.86%, 89.73%, and 89.81%, respectively. The performance of "MJP-Law" and "MJP-Law_law" in legal prediction is the same, which is 96.77%, 89.68%, 88.67% and 89.09% respectively. The performance of "MJP-Law_SG" and "MJP" in legal prediction is the same, which is 96.26%, 89.15%, 88.16% and 88.58%, respectively.

After removing the sentencing guidelines and statutory provisions, the macro F1 values of the MJP-Law model decreased by 0.51% and 0.59%, respectively. In contrast to the single MJP model, the MJP-Law model has superior predictive accuracy on low-frequency offenses.

References

[1] Medvedeva, M., Üstün, A., Xu, X., Vols, M., & Wieling, M. (2021). Automatic judgement forecasting for pending applications of the European Court of Human Rights. In Proceedings of the Fifth Workshop on Automatec Semantic Analysis of Information in Legal Text (ASAIL 2021) (pp. 12-23). CEUR Workshop Proceedings.

[2] Quteishat, E. M. A., Qtaishat, A., Quteishat, A. M. A., Al Adwan, M. A. S., Younis, M. A., & Kumar, M. (2024). Predictive Modelling in Legal Decision-Making: Leveraging Machine Learning for Forecasting Legal Outcomes. Journal of Electrical Systems, 20(3), 2060-2071.

[3] Anantathanavit, R., Chongthanakorn, J., Kongsantinart, W., Praiwattana, P., & Thaipisutikul, T. (2024, February). Utilizing AI and Natural Language Processing to Predict Supreme Court Decisions in Thailand. In 2024 16th International Conference on Knowledge and Smart Technology (KST) (pp. 45-50). IEEE.

[4] Ullah, A., Asghar, M. Z., Habib, A., Aleem, S., Kundi, F. M., & Khattak, A. M. (2020). Optimizing the efficiency of machine learning techniques. In Big Data and Security: First International Conference, ICBDS 2019, Nanjing, China, December 20–22, 2019, Revised Selected Papers 1 (pp. 553-567). Springer Singapore.

[5] Bhilare, P., Parab, N., Soni, N., & Thakur, B. (2019). Predicting outcome of judicial cases and analysis using machine learning. Int Res J Eng Technol (IRJET), 6, 326-330.

[6] Wang, Y., Gao, J., & Chen, J. (2020, October). Deep learning algorithm for judicial judgment prediction based on BERT. In 2020 5th International Conference on Computing, Communication and Security (ICCCS) (pp. 1-6). IEEE.

[7] Zhong, H., Wang, Y., Tu, C., Zhang, T., Liu, Z., & Sun, M. (2020, April). Iteratively questioning and answering for interpretable legal judgment prediction. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 01, pp. 1250-1257).

[8] Varshini, S. H., Varma, G. S., Pati, P. B., Prabhakar, P., & Parida, S. (2024). AI in the Courtroom: Enhancing Legal Decision-Making through Predictive Modelling. Journal of Information & Knowledge Management, 2450099.

[9] Alghazzawi, D., Bamasag, O., Albeshri, A., Sana, I., Ullah, H., & Asghar, M. Z. (2022). Efficient prediction of court judgments using an LSTM+ CNN neural network model with an optimal feature set. Mathematics, 10(5), 683.

[10] Strickson, B., & De La Iglesia, B. (2020, March). Legal judgement prediction for UK courts. In Proceedings

of the 3rd International Conference on Information Science and Systems (pp. 204-209).

[11] Kaur, A., & Bozic, B. (2019, July). Convolutional Neural Network-based Automatic Prediction of Judgments of the European Court of Human Rights. In AICS (pp. 458-469).

[12] Li, S., Zhang, H., Ye, L., Guo, X., & Fang, B. (2019). Mann: A multichannel attentive neural network for legal judgment prediction. IEEE Access, 7, 151144-151155.

[13] Sun, R. D., Chang, C. H., & Chien, K. C. (2023). New Horizons of Legal Judgement Predication via Multi-Task Learning and LoRA. In Legal Knowledge and Information Systems (pp. 207-216). IOS Press.

[14] Shelar, A., & Moharir, M. (2024). Judgment prediction from legal documents using Texas wolf optimization based deep BiLSTM model. Intelligent Decision Technologies, 18(2), 1557-1576.

[15] Zahir, J. (2023). Prediction of court decision from a rabic documents using deep learning. Expert Systems, 40(6), e13236.

[16] Rosili, N. A. K., Zakaria, N. H., Hassan, R., Kasim, S., Rose, F. Z. C., & Sutikno, T. (2021). A systematic literature review of machine learning methods in predicting court decisions. IAES International Journal of Artificial Intelligence, 10(4), 1091.

[17] 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).

[18] Madambakam, P., & Rajmohan, S. (2022, November). A study on legal judgment prediction using deep learning techniques. In 2022 IEEE Silchar Subsection Conference (SILCON) (pp. 1-6). IEEE.

[19] Dharma, P. Y., & Pratama, A. R. (2023, November). Legal Judgment Prediction: A Systematic Literature Review. In 2023 International Conference on Advanced Mechatronics, Intelligent Manufacture and Industrial Automation (ICAMIMIA) (pp. 691-696). IEEE.

[20] Zhu, K., Guo, R., Hu, W., Li, Z., & Li, Y. (2020). Legal judgment prediction based on multiclass information fusion. Complexity, 2020(1), 3089189.

[21] Li, H., Cai, S., & Ming, Z. (2023, October). Legal Judgment Prediction Incorporating Guiding Cases Matching. In CCF International Conference on Natural Language Processing and Chinese Computing (pp. 511-523). Cham: Springer Nature Switzerland.

[22] Iniyan, S., Raghuvanshi, N., & Maurya, P. (2024, April). Court Judgment Prediction for Article6 Using Machine Learning. In 2024 Ninth International Conference on Science Technology Engineering and Mathematics (ICONSTEM) (pp. 1-6). IEEE.

[23] Shirsat, K., Keni, A., Chavan, P., & Gosavi, M. (2021). Legal judgement prediction system. International Research Journal of Engineering and Technology (IRJET), 8.

[24] Feng, Y., Li, C., & Ng, V. (2022, July). Legal Judgment Prediction: A Survey of the State of the Art. In IJCAI (pp. 5461-5469).

[25] Wu, Y., Liu, Y., Lu, W., Zhang, Y., Feng, J., Sun, C., ... & Kuang, K. (2022, December). Towards interactivity and interpretability: A rationale-based legal judgment prediction framework. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (pp. 4787-4799).

[26] Singh, R., Kumar, B. V., Rawat, K. K., Akram, S. V., Kathuria, S., & Pachouri, V. (2023, September). Intervention of AI in Prediction of Judgement of Cases. In 2023 4th International Conference on Smart Electronics and Communication (ICOSEC) (pp. 1268-1272). IEEE.

[27] Zhao, L., Yue, L., An, Y., Liu, Y., Zhang, K., He, W., ... & Liu, Q. (2021). Legal judgment prediction with multiple perspectives on civil cases. In Artificial Intelligence: First CAAI International Conference, CICAI 2021, Hangzhou, China, June 5–6, 2021, Proceedings, Part I 1 (pp. 712-723). Springer International Publishing.

[28] Umamaheswari, S., Kanimozhi, J., & Suhashini, R. (2023, June). Building accurate legal case outcome prediction models. In 2023 2nd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA) (pp. 1-6). IEEE.

[29] Zhang, H., & Dou, Z. (2023, August). Case Retrieval for Legal Judgment Prediction in Legal Artificial Intelligence. In China National Conference on Chinese Computational Linguistics (pp. 434-448). Singapore: Springer Nature Singapore.

[30] Xuemeng Tian,Yikai Guo,Bin Ge,Xiaoguang Yuan,Hang Zhang,Yuting Yang... & Guozheng Li. (2024). Agent-DA: Enhancing low-resource event extraction with collaborative multi-agent data augmentation. Knowledge-Based Systems112625-112625.

[31] Shaobo Li,Chengjie Sun,Bingquan Liu,Yuanchao Liu & Zhenzhou Ji. (2023). Modeling Extractive Question Answering Using Encoder-Decoder Models with Constrained Decoding and Evaluation-Based Reinforcement Learning. Mathematics(7).

[32] Xiao Chaojun,Hu Xueyu,Liu Zhiyuan,Tu Cunchao & Sun Maosong. (2021). Lawformer: A pre-trained language model for Chinese legal long documents. AI Open79-84.