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

Combinatorial Press

The Role and Optimization of Multi-task Learning Models in Legal Decision Prediction

Xiongwen Wang[1,] ✉

[1] *School of Law, Dongguan City University, Dongguan, Guangdong, 523419, China*

ABSTRACT

Prediction of legal decisions using machine learning and artificial intelligence techniques has gradually become an important part of smart court technology. In addition the crime prediction and law recommendation also face the problem of easily confusing crimes. In order to solve these problems, this paper unites multi-task learning models and proposes a model fusion legal verdict prediction model. An attention neural network fusing Transformer Encoder and DPCNN encodes the key semantic information in the case description. The TF-IDF algorithm and TextRank algorithm are applied to extract the keywords of the charge, and the forward propagation network is used as a classifier to constitute a multi-task learning legal verdict prediction model. Using $CAIL2018^{9}$ legal datasets as experimental data, the metrics performance of the multi-task learning legal judgment prediction model proposed in this paper is measured on three subtasks (offense prediction, legal provision prediction, and punishment duration prediction) in LJP. Combining real case information for legal verdict prediction as well as charge differentiation. The verdict prediction results on the CAIL-Big-Multi dataset show that the mean MP value of the comparison algorithms is 82.925% in the charge prediction. And the MP index of the charge prediction of the multitask learning legal verdict prediction model proposed in this paper is 89.13%, which is significantly higher than the mean value of the comparison algorithms. And the multitask learning model incorporating the keyword information of charges in case analysis can effectively solve the problem of confusing charges.

Keywords: multitask learning model, legal judgment prediction, TF-IDF algorithm, DPCNN, attention mechanism, offense keywords

1. Introduction

The explosive growth in the number of cases not only increases the workload of judicial officers, but also increases the periodicity of the judicial process, thus increasing public anxiety. In addition, due to the differences in the knowledge base of judicial officers and the discretionary power possessed by

✉ Corresponding author.
E-mail address: herow309@163.com (X. Wang).
Received 05 January 2024; Revised 12 February 2024; Accepted 15 November 2024; Published Online 15 April 2025.
DOI: 10.61091/jcmcc127a-178

judges, as well as the subjectivity of professionals involved in a case and the influence of other information such as the place and time of the crime on the analysis and judgment of the case [1-4], the judgment results of the same case by different legal practitioners often differ and even cause controversy. Therefore, it makes sense to efficiently and accurately make judgment predictions for adjudication documents [5-6].

Legal judgment prediction is an application of natural language processing technology in the field of law, which can improve the efficiency of legal professionals, provide more professional legal advice, avoid subjective interference factors, and achieve a certain degree of judicial justice and openness [7-10]. For the public, it can widely understand the legal knowledge, clear the case situation, and obtain certain legal guidance. As one of the emerging smart court technologies, legal judgment prediction based on case description text has increasingly attracted the attention of the natural language processing community [11-13]. Charge prediction and law recommendation are two important subtasks of legal judgment prediction. These two subtasks are closely related and interact with each other, but they are often treated as independent tasks. Crime prediction and law recommendation also face the problem of confusing charges, and multi-task learning models can optimize and solve these problems [14-17].

Literature [18] describes legal verdict prediction in detail and proposes gated hierarchical multi-task learning network gei-dap, which is used in multiple subtasks in joint modeling legal verdict prediction, and based on the evaluation experiments in the subtasks, it is obtained that gei-dap is more superior to other baselines in legal verdict prediction. Literature [19] investigates the factors such as deep learning, AI environment technology for legal verdict prediction algorithms in recent years and mentions a taxonomy to categorize the collected papers into different channels and re-categorize them based on predictive techniques. Literature [20] describes a multi-task learning taxonomy based on CNN-BiGRU model that lies in improving the accuracy of legal verdict prediction. It is pointed out that CNN-bigru has good extraction ability, higher accuracy than CAIL2018-Small dataset, and improved interpretability and generalization, which is used for better utility in legal verdict prediction. Literature [21] mentions TML, a topology-aware multi-task learning framework, which proves the effectiveness of TML by extracting case features, predicting multiple civil justice subtasks and establishing connections between the tasks, and conducting experiments on a real dataset. Literature [22] introduces a multitask legal judgment prediction model that combines allegation severity subtasks. The model is not only able to obtain the corresponding caveats of punishment and the corresponding punishment terms, but also incorporates the location of the defendant and is able to keep focusing on the defendant's contextual information, and the excellent performance of the model is verified through experiments on the CALL2018 dataset. Literature [23] proposes a hierarchical dependency-based legal judgment prediction framework, which perfectly integrates task judicial logic, labeling topological relations, and hierarchical semantic knowledge to make its impact on judgment more credible and interpretable. Its effectiveness has also been demonstrated in experiments and its performance due to existing prediction methods. Literature [24] briefly describes legal judgment prediction, proposing an approach based on multi-task learning and focusing on the logical relationships between its sub-tasks, aiming to examine the associations that exist between the tasks in order to guide the model in unfolding interpretable reasoning. Its effectiveness is demonstrated in a comparative test of it with existing models. Literature [25] pointed out MTL-LJP, a legal judgment prediction algorithm optimized for multi-task learning, which uses EnMo to encode cases and takes SiMa to compute the similarity matrix between cases. And MuTa is able to modify the prediction results based on the similarity of associated cases. Experiments with real case datasets show that MTL-LJP outperforms previous methods in the legal judgment prediction subtask.

In this paper, we propose a multitask learning model for legal verdict prediction based on model fusion to address the challenges of crime prediction and law recommendation. After obtaining the case description from the word vector module, the Transformer Encoder is utilized to capture the text long-

range features to obtain a high-level representation of the case description. Then the DPCNN model is used to encode the legal text information and bring in the multi-attention mechanism to extract the key semantic information of the case description. Real legal datasets are selected as the basis for algorithm simulation, and a variety of high-performing prediction algorithms are selected for index evaluation. Multi-task learning model for the keyword information of the crime is used for confusing crime detection. And the multitask learning model for legal verdict prediction proposed in this paper is used for real case prediction.

2. Joint multitask learning for prediction of legal judgments

2.1. Predictability of Judicial Decisions in the Age of Big Data

The concept of predictability is mainly found in the fields of statistical physics, mathematics, human-computer interaction and utterance analysis. The latter two fields can be said to be primarily where Big Data research is taking place.

For the purposes of this paper, if the courtroom trial process is treated as a dialog system. It usually means that progress on the level of system executability is performed in a pre-established and defined way, and that the time of execution can be predicted. If judicial decisions are treated as datatable text or speech, they contain nothing more than what one would expect.

In general, the predictability of judicial decisions in the context of this paper is the predictability of the specific outcome of the judge's decision (including the outcome of litigation in civil and administrative proceedings and the outcome of conviction and sentencing in criminal proceedings).

2.1.1. Significance of predictability. Advancing the predictability of judicial decisions has the following positive implications:

First, judges can save time in the integration of evidence and the collection of resources, so that they can invest more energy in promoting the argumentation of judicial decisions.

Second, for the legal people, especially lawyers, in terms of whether to receive a particular case, to provide the probability of success and the difficulty of evidence collection and other aspects of the information, so as to help them to make both for the legal people and the parties are the most efficient and fair decision-making.

Third, it helps to promote the science of evidence collection and the progress of information technology into judicial argumentation.

Fourth, it helps the legal practitioners to effectively target the possible litigation strategies of the opposing party based on the prediction of the antecedent, so as to confirm the planning and response of their own litigation argumentation.

Fifth, in terms of expanding data sources, it helps to promote the computerization of moot court and similar past case results.

Sixth, and most importantly, the predictability of judicial decisions corresponds to the stability of the rule of law.

2.1.2. Sentence forecasting process. The Legal Judgment Prediction Process (LJP) proposed in this paper is mainly composed of three parts: data collection, data preprocessing and judgment result prediction.

1) Data Collection. The experimental data for the study of legal judgment prediction based on the big data of legal judgment documents consists of three parts, namely, the judgment documents network, the encyclopedic knowledge base, and the related judicial class review dataset, respectively.

2) Data Preprocessing. Different kinds of data, such as legal documents, are normalized into the data format of judicial evaluation, and the factual descriptions, relevant legal articles, crimes involved, and sentence length of each case in the data are extracted. Then extract the information of the elements involved in the factual descriptions of different types of cases according to the relevant laws and

regulations, such as “having children after marriage” in divorce cases, “the existence of a labor relationship” in labor cases, “a written repayment promise” in loan cases, and so on. These elemental information is the key information that affects the judgment result, and is also the input information of the legal judgment prediction model. This paper extracts the elemental sentences in the description of case facts based on the judgment element extraction model, and combines the elemental information of each case with the relevant laws, crimes and sentences of the case to get the data sets of three sub-tasks of legal judgment prediction.

3) Sentence prediction. In this paper, using the preprocessed large-scale dataset, based on the deep learning method, the law prediction model, crime prediction model and sentence prediction model are constructed respectively, and the three types of models are used to make law prediction, crime prediction and sentence prediction for the case respectively, and finally get the sentence prediction results of the case.

2.2. Problem description and model overview

To address the challenges of charge prediction and law recommendation, this paper proposes a multitask learning model for legal verdict prediction based on model fusion [26-27]. With $X = \left\{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\right\}$ denoting all case description texts, where N is the number of cases. Also $y^i = \left\{y_1^i, y_2^i, \ldots, y_C^i\right\}$ is x^i the corresponding offense label, where C is the number of offense labels, and each $y_j^i \in \{0,1\}$ indicates y_j^i whether the offense is violated or not. Similarly, $z^i = \left\{z_1^i, z_2^i, \ldots, z_L^i\right\}$ is x^i the corresponding law label, where L is the number of law labels and each $z_j^l \in \{0,1\}$ indicates z_j^i whether the law has been violated.

The goal of the model is to predict, given a case description $x^i = \left\{w_1^i, w_2^i, \ldots, w_m^i\right\}$, where m is the length of that case description, the charge $\hat{y}^i$ of its defendant and the related legal articles $\hat{z}^i$ involved, where $\hat{y}^i \in R^C$, $\hat{z}^i \in R^L$.

The multitask learning model for legal verdict prediction is based on an encoder-decoder framework for the joint construction of a multitask learning model for the tasks of crime prediction and law article recommendation. The coding end of the overall network structure of the model is mainly composed of three parts. They are case key information coding module, charge keyword coding module, and global case information coding module.

2.3. Module for coding key case information

In this subsection, an attention neural network incorporating Transformer Encoder and Deep Pyramidal Convolutional Neural Network (DPCNN) is proposed to encode the key semantic information in the case description.

The structure of the case key information encoding module is shown in Fig. 1, in which the case description is obtained from the word vector module and then encoded by Transformer Encoder to obtain the case description representation. At the same time, a simple and effective method is used to extract the relevant K legal articles from the case description, which is inputted into the DPCNN to encode the relevant legal articles to obtain the relevant legal article representation. The key semantic information in the case description is obtained by using the attention mechanism to combine the relevant law article representation and the case description representation.

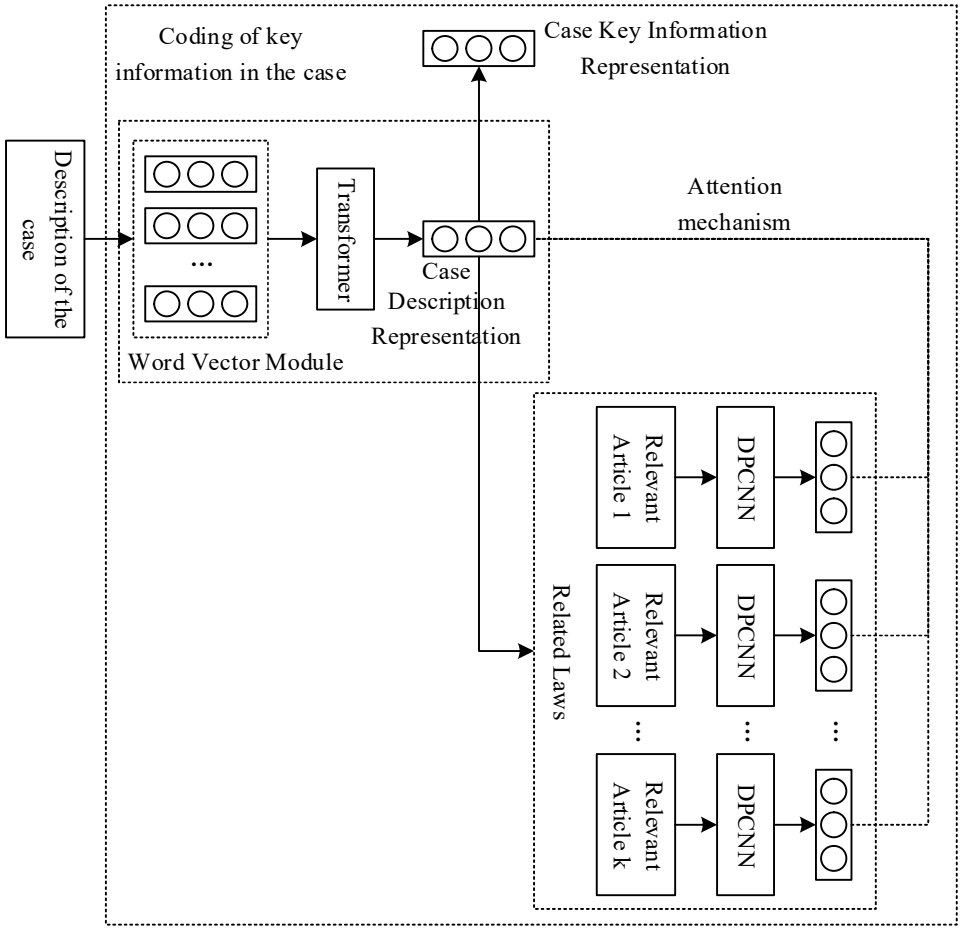


Fig. 1. The structure of the key information encoding module of the case

2.3.1. Transformer Encoder-based encoding of case descriptions. In this paper, we use Gensim's word2vec open-source tool to set parameters such as training mode, dimension of output word vectors, window size, minimum word frequency, sampling frequency, etc., to train the text of the case description after segmentation in order to obtain word vectors.

After obtaining the word vector $x^i = \left\{d_1^i, d_2^i, \ldots, d_m^i\right\}$ for a given case description from the Word Vector module, where m is the length of the given sequence and the word vector dimension is d_f. A high-level representation of the case description is obtained by using a Transformer Encoder to capture the long distance features of the text. Transformer Encoder is composed of a multi-head self-attention mechanism and a feed-forward neural network, while residual connectivity and layer normalization are used between the two sub-modules [28].

The multi-head attention used in this paper takes inputs Q_x, K_x, and V_x and first puts them through h sets of linear transformations. Then the result is fed into the deflated dot product attention, and the results of h times of deflated dot product attention are spliced. The value after applying another linear transformation to it is the result of the multi-head attention, which is calculated in Equation (1):

$$MultiHead\left(Q_x, K_x, V_x\right) = Concat\left(head_1, \cdots, head_h\right)W^o \tag{1}$$

where $Q_x = K_x = V_x \in R^{m \times d_f}$ is a representation of the word vector matrix X with positional information. The result of the i nd deflationary dot product attention is denoted as $head_i = Attention\left(Q_x W_i^Q, K_x W_i^K, V_x W_i^V\right)$. The linear transformation weight matrices W_i^Q, W_i^K, and

W_i^V all have dimension $d_f \times d_k$.

The general formula for normalization is shown in equation (2), where μ is the translation parameter, σ is the scaling parameter, b is the re-translation parameter, g is the re-scaling parameter, and the final data obtained conforms to a distribution with mean b variance g^2. The different methods of normalization are mainly the differences in the method of solving for μ and σ. That is:

$$h = f\left(g \cdot \frac{x-\mu}{\sigma} + b\right) \tag{2}$$

And LN is a proposed way to normalize data in response to the shortcomings of batch normalization (BN). The operation of layer normalization is shown in Eq. (3), where X is the word vector matrix with positional coding, Z is the output matrix of multi-headed self-attention, and Z and Z' have the same dimension. Namely:

$$Z' = LN\left(X + Z\right) \tag{3}$$

In each sublayer of the Transformer Encoder, a fully connected feed-forward neural network (FFN) is accessed after the multi-head self-attention, as described in Equation (4):

$$FFN = \max\left(0, ZW_1 + b_1\right)W_2 + b_2 \tag{4}$$

where Z' is the output of the normalized multi-head attention with dimension $m \times d_f$, and the parameter matrices $W_1 \in R^{d_{ffn} \times d_{fin}}$ and $W_2 \in R^{d_{ffn} \times d_f}$, d_{ffn} are the dimensions of the feed-forward neural network, so that the dimension of the FFN output remains $m \times d_f$.

2.3.2. Extraction and coding of relevant laws and regulations. The given case description text is encoded by Transformer Encoder to obtain a high-level case description representation e^j with dimension $m \times d_f$. In this paper, a simple affine transformation is applied on the basis of case description representation e^j, and then a sigmoid function is used to compute the score of each law article, which is calculated in equation (5):

$$score_{art} = sigmoid\left(W^s e^i + b^s\right) \tag{5}$$

where W^s and b^s are the weight matrix and bias vector. In the training phase, this paper uses real relevant law labels for training, and in the prediction phase, the top k relevant laws are taken as the candidate relevant laws for a given case description according to the score ranking.

Through the extraction phase of relevant laws, k relevant laws are obtained for each given case description, and the text information of these relevant laws is encoded using the DPCNN model to obtain their relevant law representations.

DPCNN in order to simplify the training of the deep network, in the convolutional layer using a pre-activation method, through two layers of convolutional operation of the formula shown in (6), where $\sigma(\cdot)$ is a nonlinear activation function, here take the RELU function, W_1^{dp}, b_1^{dp} and W_2^{dp}, b_2^{dp} are the parameters of the first two layers of convolutional layer, respectively. Namely:

$$f\left(a^j\right) = W_2^{dp}\sigma\left(W_1^{dp}\sigma\left(a^j\right) + b_1^{dp}\right) + b_2^{dp} \tag{6}$$

DPCNN uses the additive operation $a^j + f\left(a^j\right)$ for residual concatenation in order to achieve constant mapping, which greatly alleviates the problem of vanishing gradients and makes training of deep networks possible.

2.3.3. Attention mechanism-based extraction of key information of the case. Like most Seq2Seq models, Transformer uses an encoder-decoder framework. Where the encoder consists of six identical encoder modules stacked on top of each other, and each encoder module consists of two sub-modules. Multihead Attention Layer and Feedforward Neural Network, respectively. Both sub-modules add residual linking and layer normalization, and let the input be x. The outputs of the two sub-modules are given in Eq:

$$output_{Att} = LayerNorm\left(x + Attention\left(x\right)\right) \tag{7}$$

$$output = LayerNorm\left(output_{Att} + FNN\left(output_{Att}\right)\right) \tag{8}$$

The Transformer decoder also consists of six identical decoder modules stacked on top of each other, and the decoder modules are similar in structure to the encoder modules. The only difference is that a masked multi-attention layer is added to the bottom layer of the decoder module, so that the decoder will not be exposed to future information during training by masking the weight of the attention.

The multi-head attention mechanism in Transformer consists of deflated dot product attention. The deflated dot product attention is similar to the common attention mechanism that uses dot products to compute similarity, except that a deflation factor is added. First, the dot product of query Q and key K is computed, then divided by the dimension of a single input for deflation, and then a softmax function is used to compute the weight of each attention corresponding to key K, and finally multiplied by the value V to obtain the attention representation. The formula can be expressed as follows:

$$Attention\left(Q,K,V\right) = soft\max\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{9}$$

where d_k denotes the dimension of a single key vector, and $\sqrt{d_k}$ acts as a deflation factor to regulate the inner product in the molecule, preventing it from being too large and causing the computed attention weights to be mostly too close to zero.

The multi-head attention mechanism proposed in Transformer projects Q, K, and V through several different linear transformations before calculating the attention, and performs the attention calculation in different feature dimensions separately, and then finally splices the attention calculation results of multiple heads, and maps the spliced calculation results to the output dimensions with a linear layer. The specific calculation of the multi-head attention mechanism is shown in Eqs. (10) and (11):

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{10}$$

$$MutiHead\left(Q,K,V\right) = concat\left(head_1, head_2, \ldots, head_h\right)W^o \tag{11}$$

Since Transformer discards the recurrent and convolutional mechanisms and is based exclusively on the attention mechanism and feed-forward neural networks, it is not possible to model the information about the position of the words in a sentence. In order to make it possible to utilize the word order information of the sequence, the authors introduced positional coding, where the ind element of the position vector for an input with position pos is computed as follows:

$$PE\left(pos, 2i\right) = \sin\left(pos/10000^{2i/d_{model}}\right) \tag{12}$$

$$PE\left(pos, 2i+1\right) = \cos\left(pos/10000^{2i/d_{model}}\right) \tag{13}$$

where d_{model} is the dimension of the model word vector. The position vectors and word vectors are directly summed as inputs to the model.

There are two main advantages for Transformer to abandon the traditional RNN and CNN and rely

only on the attention mechanism for feature extraction of text context features. One is that the computation of the attention mechanism does not have a before-and-after computation dependency like RNN, which makes the model parallelizable and greatly improves the computational efficiency. Another point is that the computation of attention ignores the distance between different words in the structure of the model, which allows better modeling of long-distance dependent features, which makes the basic module of Transformer can be stacked in multiple layers and will not be affected by the loss of information over long distances. These two advantages made Transformer the base component of many later large-scale pre-trained language models.

2.4. Multi-task learning model incorporating charge keyword information

2.4.1. Multi-task learning. Multi-task learning (MTL) is a learning approach in which several related tasks are learned together based on a shared representation. As opposed to single-task learning, the main feature of this learning approach is to learn a number of related tasks in parallel, with the tasks helping each other to learn by means of underlying sharing, and the main goal is to improve the generalization ability of the model.

The single-task learning model tends not to consider the correlation between problems, so dealing with problems with independent single tasks will create a series of problems. At this time, the multi-task learning model that puts a number of related tasks together to learn comes into being, and multiple related tasks share what they have learned to achieve complementary information, an advantage that single-task learning does not have. In terms of generalization ability, correlated multitask learning is more effective than single-task learning.

Multi-task learning models can be divided according to whether the correlation between tasks is examined or not, one class of models does not consider the logical dependency between multiple tasks, and only adopts the shared structure and outputs the prediction results at the same time, e.g., MMoE. Another class of models in which there is some kind of logical correlation between multiple tasks, and the prediction results of one task need to rely on the prediction of the other tasks, e.g., TOPJUDGE.

TOPJUDGE is a multitask model of legal adjudication based on a topological order structure. The model references the adjudication logic of human judgment and incorporates the dependency of multiple subtasks in judgment prediction, i.e., the prediction result of a particular task is affected by the prediction results of the tasks on which it depends.

The model assumes that the relationships of legal judgment prediction tasks form a network structure, i.e., the list of tasks T should satisfy topological constraints:

$$t_i \square\ t_j \tag{14}$$

where $t_i \left(i = 1, 2, \ldots, n\right)$ denotes the i nd subtask and n denotes the total number of subtasks. Eq. (14) denotes that the j th subtask depends on the prediction result of the i th subtask. $D_j = \left\{t_i \middle| t_i \square\ t_j\right\}$ then denotes the set of all tasks on which the j th subtask depends.

2.4.2. Crime keyword extraction and incorporation. In this paper, 2 unsupervised methods are used to extract a number of keywords for each charge in a large number of unlabeled texts to construct a keyword list of charges. These 2 methods utilize TF-IDF and TextRank to rank the scores of keyword candidates, respectively. When the charge keyword list is constructed, the charges are classified using binary classification. The SVM classifier is utilized to obtain the charge keywords related to the description of the case, which are further incorporated into the multi-task learning model above. Here, vector summation is used to encode the keywords, which are combined with the case encoding.

1) TF-IDF, TextRank Based Keyword Extraction for Charges. In order to assess the importance of a word to a document, the TF-IDF algorithm is proposed, the basic guiding principle of the algorithm is that the importance of a word increases proportionally with the number of times it appears in the

current document, but at the same time decreases inversely with the number of times it appears in all documents. That is, a word appears more often in a document, but at the same time in all the documents appear less often, the more representative of the importance of the word in the document.

TextRank algorithm is a keyword extraction algorithm based on word graph model. Firstly, it is considered that the co-occurrence between words represents a recommendation relationship. That is, the words that co-occur with important words are also important, according to this relationship between words to construct a network graph without direction and right edges. Then iterate on the graph to get the importance ranking of the words in the document, and finally regard the keyword extraction problem as the importance ranking problem of the words that constitute the document.

2) Encoding and incorporation of offense keywords. Incorporate the information of the law in the neural network model, and use the binary classification method to classify the law and obtain K law. Inspired by it, the same binary classification is used to classify the charges. Utilizing SVM as a classifier, the case description is taken as an input, and K counts related to the case description are output.

Further the offense keyword $\{k_1, k_2, \cdots k_m\}$ is obtained from the constructed offense keyword list and its vector representation is given as $\{e_{k_1}, e_{k_2}, \cdots e_{k_m}\}$. The keyword is then encoded using vector summation to get the vector representation of the keyword h_k. It is then vector spliced with the case encoding as the final vector representation at the encoding end.

2.5. Vector Splicing and Judgment Prediction

In the fusion of artificial intelligence technology and judicial technology, there is a natural interdependence between law prediction, offense prediction and sentence prediction. In reality when the judge is judging a case, he usually finds the legal basis related to the case first. Then they judge the case according to the relevant legal provisions, and then combine various evidence and facts to convict (sentence, fine) the case.

In this study, recurrent neural networks are not used as the final classification method and there is no intersection of any parameters between the sub-task classifiers. The rationale for this is that the required information about the case has been segmented after encoding the information about the case before classification. This means that there is no need to process the sequence of vectors, only the individual vector information required for each subtask. Therefore, no practical gain will be obtained if a recurrent neural network is used. Based on this, the basic component of the classifier is the forward propagation network and the corresponding architecture of the classifier is shown in Fig. 2.

In the figure, the law keyword vector F is spliced with the separated vector T_1 in the main encoder, after which it enters the classifier to realize the prediction of the law, and in the relationship of the sub-tasks, reference is made to the process of the judge's judgment in reality, based on the judgment process of firstly predicting the law, then determining the crime based on the law, and finally determining the sentence. The formula for law prediction is shown below:

$$T_i = soft\max\left(\mathrm{Re}lu\left(T_l^1 W_l^1 + b_l^1\right)\right) \tag{15}$$

$$T_l^1 = \mathrm{Re}lu\left(T_m^1 W_m^1 + b_m^1\right) \tag{16}$$

The structure of crime and sentence prediction is basically the same as that of law prediction, but the difference is that the input vector needs to be changed according to the different objectives. For example, the input vectors of the charge prediction include variables F, T_2, and the intermediate vector of the statute prediction, T_l^1. The input vectors of the sentence prediction include T_d, T_3, T_l^1 and the intermediate vector of the charge prediction, T_{ck}^1. Finally, we get T_l, T_{ch}, and T_p. The three vectors obtained from the Softmax function are in fact the probability distributions of the categories of the intelligent judicial sentence prediction, and the experimental expectation is that the value of correctly labeled positions is 1, and all other values are 0. All other values are 0.

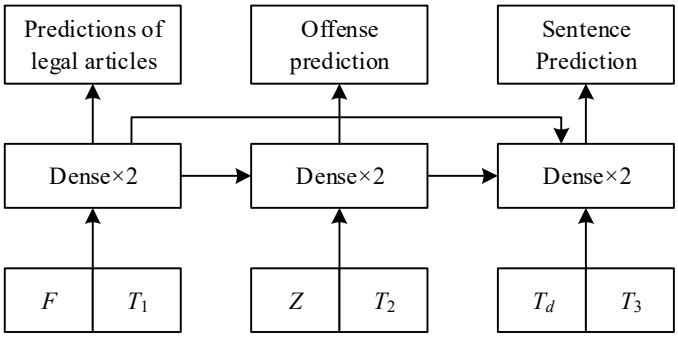


Fig. 2. The structure of the classifier

3. Experimentation and analysis

3.1. Data sets

In order to demonstrate the effectiveness of the multi-task learning legal judgment prediction model proposed in this paper, extensive experiments were conducted using the $CAIL2018^{9}$ dataset. This dataset consists of two sub-datasets, CAIL-Small and CAIL-Big.

After processing the above datasets, this paper randomly divides CAIL-Small and CAIL-Big-Multi into two parts, respectively, of which 80% is used as the training set and the rest is used as the test set (class size distribution holds). As a result, the detailed statistics of the final experimental dataset are shown in Table 1.

Table 1. Statistics of experimental data sets

Data set	Data set segmentation	Case	Offence	Law	Penalty period
CAIL-Small	Training	150174	185	156	10
	Testing	36965	185	156	10
CAIL-Big-Multi	Training	49413	104	99	10
	Testing	12715	101	99	10

3.2. Experimental setup

In this paper, a customized law-related dictionary is used and then Jieba is used to slice and dice all the case description words on CAIL-Small and CAIL-Big-Multi. Also, this paper sets the maximum sentence length to 120 words and the maximum document length to 25 sentences.

For case descriptions, this paper applies Word2Vec to generate pre-trained word vector models and sets the word embedding size and vocabulary size to 500 and 90,000, respectively.For legal concepts and penalty descriptions, the total vocabulary size is the number of their keywords and the embedding size is set to 100 and 15, respectively. In the word encoder module, the Bi-LSTM is set to a single layer and the hidden cell size is set to 256. In the sentence encoder module, the Transformer encoder is set to single layer, the attention head is set to 8, and the dropout probability is set to 0.5. In this paper, we also set the batch size to 128 for all models, where each model is trained for 50 rounds and the final model is evaluated on a test set.

In this paper, the weight factor of each subtask is set λ_i to 1, and the Adam optimizer with a learning rate of 0.003 is used to minimize the loss function of the multitask-learning legal judgment prediction model.

3.3. Contrasting models

In order to demonstrate the effectiveness of the framework, the multi-task learning framework for

legal judgment prediction modeling proposed in this paper is compared with the following methods:

CNN: a CNN-based text categorization model that extracts text features through multiple convolutional kernels.

GRU: a gated recurrent unit for text categorization with better understanding of long text semantics.

BiGRU: A bidirectional gated recurrent unit based on BiGRU, which can effectively improve the effect of GRU network.

HAN: A hierarchical attention network encoded based on GRU network, which can analyze text from word level to sentence level.

Topology: a topological multi-task learning framework for LJP, which performs multi-task learning by formalizing dependencies between subtasks as directed acyclic graphs.

MPBFN-WCA: This method is a multi-perspective dual feedback network that uses the word collocation attention mechanism and exploits the topology between subtasks.

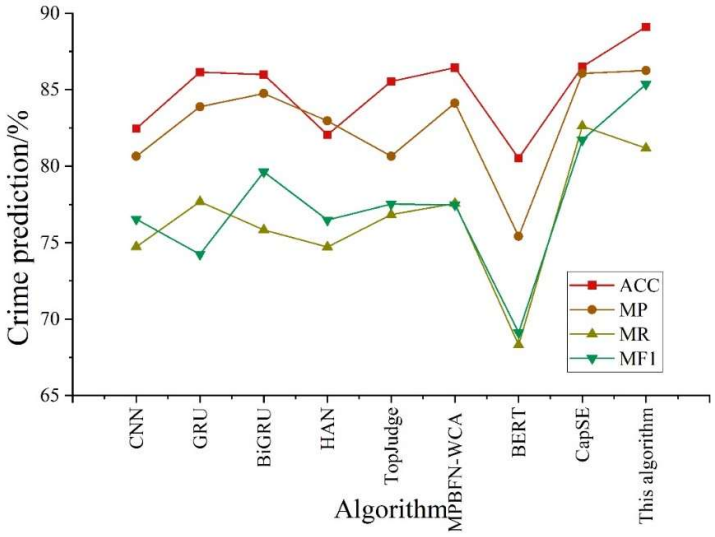
BERT: BERT is a migration learning model pre-trained in a self-supervised manner on large data corpora.

CapSE: A capsule network-based sentence encoder for LJP.

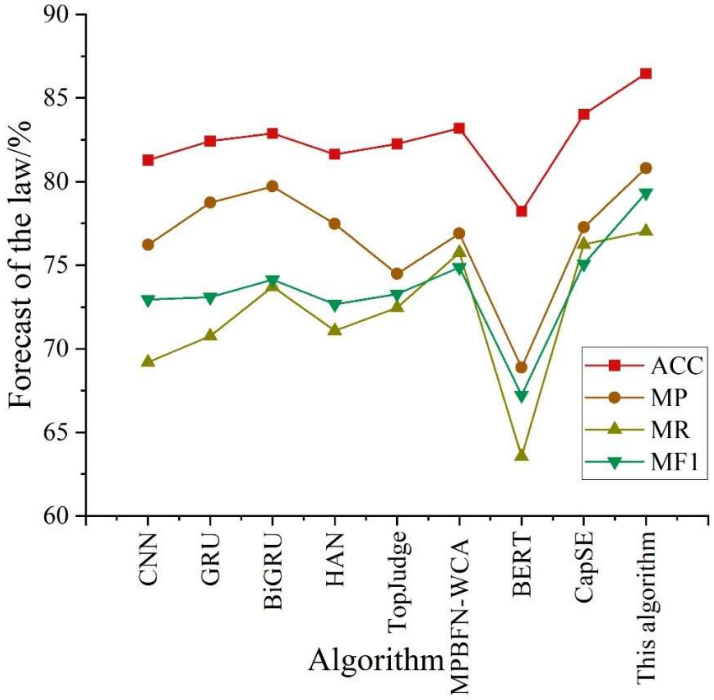
3.4. Overall Performance Comparison

For the three subtasks in LJP, i.e., charge prediction, legal provision prediction, and punishment duration prediction, the verdict prediction results of each algorithm on CAIL-Small are shown in Fig. 3, and Figs. (a) to (c) show the charge prediction results, the legal provision prediction results, and the punishment duration prediction results, respectively.

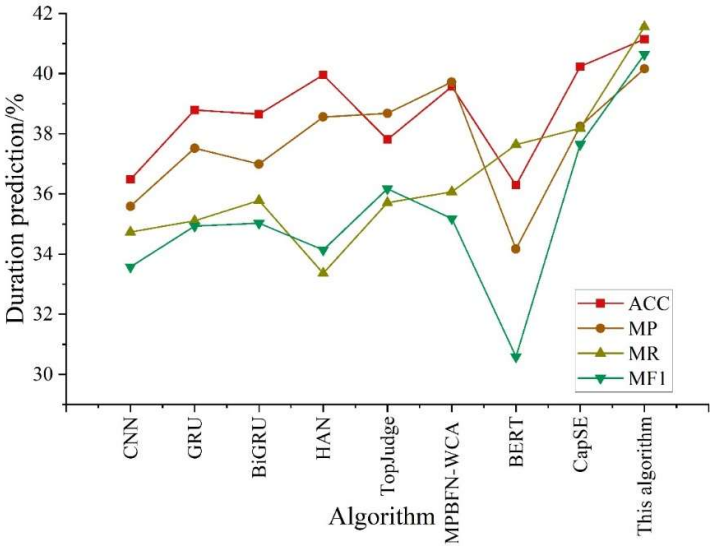
The multi-task learning legal verdict prediction model proposed in this paper is higher than other comparative algorithms in the three subtasks of crime prediction, legal text prediction, and punishment duration prediction. The prediction means of the multitask learning legal sentence prediction model proposed in this paper are 85.47%, 80.91%, and 40.88% for the three subtasks, respectively.



(a) Crime prediction



(b) Forecast of the law



(c) Duration prediction

Fig. 3. The algorithms are predicted in the CAIL-Small

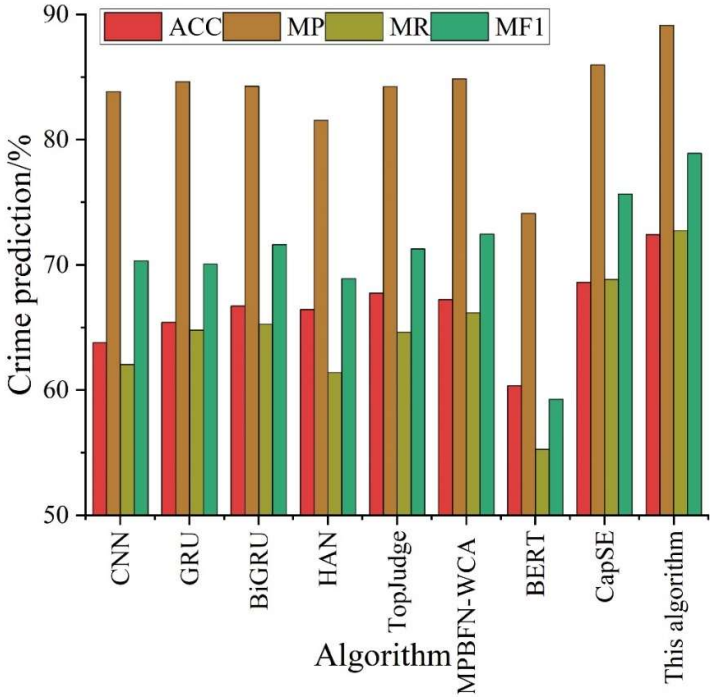
The sentence prediction results on CAIL-Big-Multi are shown in Fig. 4, and Figs. (a), (b), and (c) show the results of the charge prediction, the MP prediction, and the punishment duration prediction, respectively.

In the offense prediction, the MP prediction index of each algorithm performs better. The mean MP value of the comparison algorithms is 82.925%. The MP metric for offense prediction of the multitask learning legal sentence prediction model proposed in this paper is 89.13%, which is significantly higher than the mean value of the comparison algorithms.

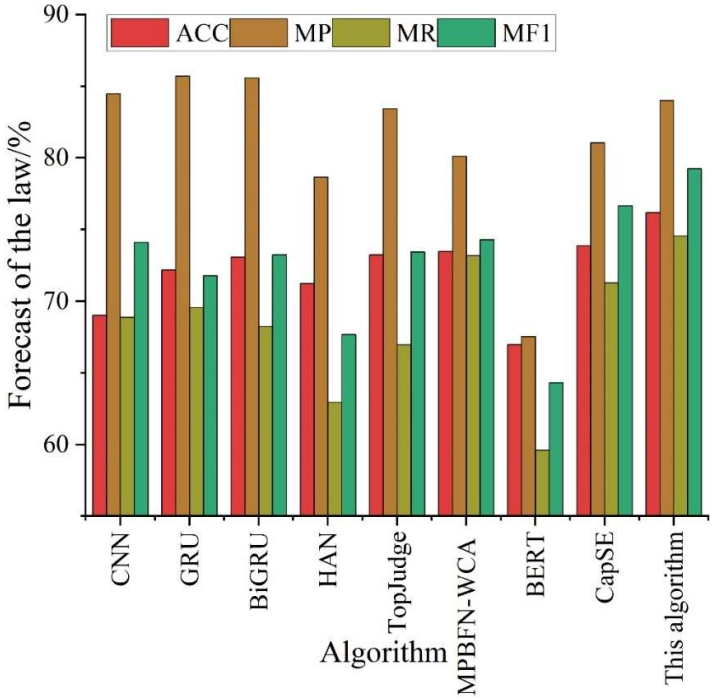
In legal provision prediction and punishment duration prediction, the prediction mean value of each indicator of this paper's algorithm is 78.485% and 38.715%, which represent the highest prediction value of the comparison algorithms, respectively.

In crime prediction, legal text prediction, and punishment term prediction, this paper's model improves 3.24%, 2.59%, and 2.42% in MF1 compared to the second best model (CapSE) on the CAIL-Big-Multi dataset, respectively. This indicates that the multitask learning legal judgment prediction model proposed in this paper is an effective LJP framework.

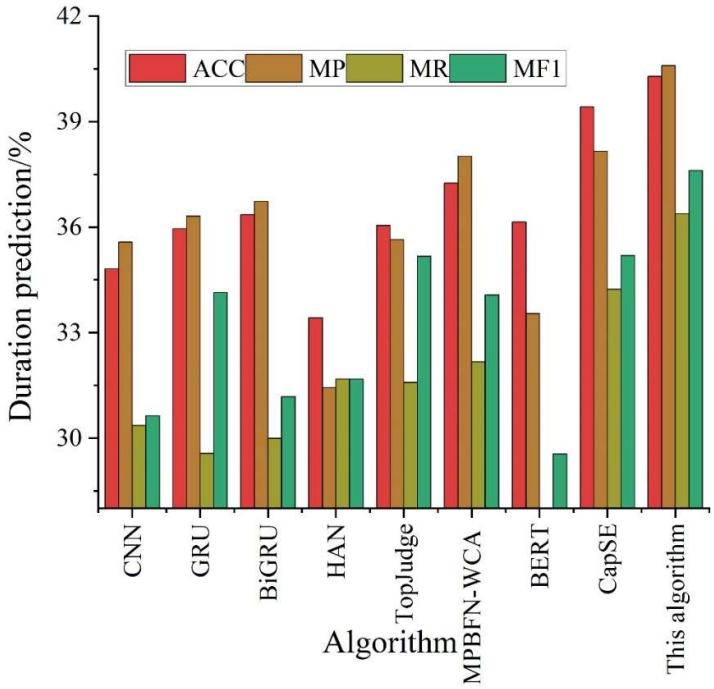
Overall, the legal judgment prediction model with multi-task learning proposed in this paper achieves the best performance in MF1 on both datasets.



(a)Crime prediction



(b)Forecast of the law



(c)Duration prediction

Fig. 4. CAIL-Big-Multi-decision prediction results

3.5. Comparative analysis of keyword coding methods

In order to further validate the performance of the multi-task learning model based on Attention encoding fusion keyword method proposed in this paper, it is necessary to compare and contrast the experimental results of different keyword encoding methods in order to verify the validity in other crime confusion and related law confusion problems.

The experimental results of combining keyword methods for legal judgment prediction are shown in Table 2.

1) From the experimental results, it can be seen that there is a big difference between the results of the judicial judgment prediction task with the addition of keyword information and without the key. Compared to the multi-task learning model PLE without considering keyword information, the relative improvement of the F1 value of the PLE model introducing keyword information based on SVM encoding in the three tasks of crime prediction, law sentence prediction and sentence prediction is 3.23%, 3.59% and 3.23%, respectively. The relative enhancement of the F1 value of the PLE model based on Attention coding fused with keyword information in the three tasks of crime prediction, related law prediction and sentence prediction is 8.38%, 8.68%, and 6.45%, respectively, which indicates that the keyword information is very important for the enhancement of the task of legal discrimination prediction.

2) After the Attention mechanism is used to encode the keyword information, compared with the PLE model based on SVM encoding fusion of keyword information, the relative enhancement of the F1 value of the PLE model based on Attention encoding fusion of keyword information in the three tasks of crime prediction, related law prediction, and sentence prediction is 5.15%, 5.09%, respectively, 3.22%, indicating that the method incorporating Attention mechanism has better performance in dealing with crime confusion as well as statute confusion problems.

Table 2. The legal judgment of the method of keywords is predicted

Task	Crime			Relevant law			Prison term		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
PLE	75.9	65.48	71.03	79.65	73.45	75.43	70.45	69.04	69.11
SVM+PLE	79.61	70.52	74.26	83.41	76.38	79.02	72.61	71.55	72.34
Attention+PLE	82.33	76.71	79.41	85.66	79.21	84.11	74.23	73.94	75.56

3.6. Treatment of Confusing Offenses

Confusable offenses are those offenses that have a high degree of definitional similarity between them, with only minor differences. They can be easily confused either when the judge decides the case or when the machine predicts it. For example, the crime of robbery and robbery. Robbery is the use of violence, coercion or other methods on the spot by the perpetrator against the owner, custodian, janitor or holder of public or private property to compel them to hand over the property immediately or to snatch the property immediately. The crime of robbery, on the other hand, means that violence is committed against an object. However, if the violence is committed against both the person and the object, it can constitute the crime of robbery. As can be seen from their definitions, the main difference is what the object of the crime is, robbery is mainly against people. And robbery is mainly against things. Due to the ambiguity that may exist in the description of the case of the adjudicatory instruments, the model will have the possibility to confuse the crime of robbery and robbery when making judgments. Therefore, an approach is needed to try to solve the problem.

In this paper, we try to extract the corresponding keywords from the case descriptions by means of keyword extraction, and after filtering out the deactivated words as well as some irrelevant words, we convert them into word vectors by word2vec, which are used as the second input of the model.

Two unsupervised keyword extraction approaches are used in this experiment for comparison, one is statistical feature-based keyword extraction (TF-IDF) and the other is word graph model-based keyword extraction (TextRank).

For the extraction of keywords for the offense, by taking the above two approaches. For each case description 100 keywords are extracted and converted to word vectors by word2vec as the second input to the model.

The test set used for the experiment is 1689 entries, and the label selection threshold is 0.5. The model is selected from the SVM which has worked well before, and the result comparison is shown in Table 3.

It can be seen that by incorporating the model after the case description keywords, the effect will be improved. Comparatively speaking, TextRank, although improved in subset accuracy, has a poor performance effect on other metrics. The experimental F1 value for keyword extraction using TF-IDF is 89.13% with an accuracy of 85.68%. The problem of confusing charges is solved by using keywords incorporated into the charges.

Table 3. Result contrast

	Hamming loss	Accuracy	F1	Label ranking loss
Unprocessing	0.00112369	0.85659	0.89693	0.156093
TextRank	0.00117445	0.85112	0.86442	0.132145
TF-IDF	0.00119368	0.85677	0.89125	0.112567

Next, the effect of modeling using TF-IDF extracted keywords as the second input was further viewed. A comparison of the results of the experimental and control groups for some of the confusable offenses is shown in Table 4.

It can be seen that the performance of the model with the addition of case description keywords is improved on several confusable offenses relative to the original unprocessed single-input model.

When unprocessed, the number of drug smuggling, trafficking, transportation, and manufacturing charges predicted to illegal drug possession charges is 89, which reaches 5.27%. And after the inclusion of case description keywords, the number of misclassifications dropped to 45, or 2.66%. The performance on several other pairs of confusing charges also improved.

Therefore, combining several evaluation indexes of the experimental group and the performance of the improved model on the confusing charges. It can be proved that the method can effectively improve the prediction error of confusable charges.

Table 4. Part of the confusion was compared to the control group

	Control group		Experimental group	
	Quantity	Rate/%	Quantity	Rate/%
Smuggling, trafficking, transport and manufacturing drugs → Illegal possession of drugs	89	5.27	45	2.66
Theft → Mugging	157	9.30	46	2.72
Pick up trouble → Intentional injury	93	5.51	87	5.15
Take bribes → Non-state workers take bribes	41	2.43	37	2.19
Take drugs → Smuggling, trafficking, transport and manufacturing drugs	123	7.28	89	5.27

3.7. Results of the legal judgment prediction experiment

In order to show more obviously the prediction effect of the above models on the crime, relevant laws and sentence, the following shows a case factual example and the prediction results of the above models under the three subtasks. The model prediction results are shown in Table 5.

The text of the case facts is: the facts of committing the crime of × × In December 2020, A (a separate case) provided accommodation for prostitutes in the leisure center he operated, and accommodated prostitutes for prostitution. Defendant B was responsible for collecting table fees at the leisure center.

At about 20:00 on February 5, 2021, when the Public Security Bureau police conducted an inspection of the leisure center, they seized prostitutes C Mou and D Mou and prostitutes X Mou and Y Mou on the spot.

After the crime, the defendants truthfully confessed to their crimes. The evidence of the above facts as determined by the public prosecution authorities include: household registration information, mediation agreement and other documentary evidence, witnesses and other people's testimonies, the victim's statement, the defendant's confession and defense, judicial appraisal opinion, identification, extraction transcripts and so on.

Related Offenses:

TAG: organizing, forcing, inducing, accommodating, procuring prostitution.

PLE: organizing, forcing, inducing, accommodating, introducing prostitution.

TOP-JUDGE: Manufacturing, trafficking, distributing obscene materials.

MMoE: Manufacture, trafficking, dissemination of obscene articles.

MoE: provoking trouble.

Shared Bottom: Intentional injury.

Related Laws:

Tag: 287. PLE: 287. TOP-JUDGE: 287. MMoE: 312. MoE: 152. Shared Bottom: 204.

Sentence: Label: 11. PLE: 11. TOP-JUDGE: 11. MMoE: 11. MoE: 5. Shared Bottom: 11.

As can be seen from the table, only the multi-task learning model based on Attention encoding fusion keyword method proposed in this study has the same predicted results and actual values on the three subtasks for this case factual text case.

Table 5. Model prediction effect

The fact text of the case	Related crime	Relevant law	Prison term
Someone's procuratorate charges: The fact that the fact of the crime is in December, 2020, a (another case treatment) in the business of the leisure center for the prostitution of women to provide accommodation, and the prostitution of prostitutes. The defendant b is responsible for charging a fee at the leisure center. On February 5, 2021, the public security police examined the leisure center of the public security bureau, and found the x and x and y of the sex of the prostitute on the spot. After the crime, the defendant faithfully confessed his crime. The evidence of the above facts of the public prosecution organ is that the certificate of the household registration, the conciliation agreement, etc., the testimony of the witnesses, the testimony of the victim, the statement of the victim, the confession of the defendant, the judgment of the judgment, the identification of the judgment, the extraction of the written record, etc.	Label: Organize, force, lure, maintain, introduce prostitution. PLE: Organize, force, lure, maintain, introduce prostitution. TOP-JUDGE: To manufacture, sell and disseminate pornographic materials. MMoE: To manufacture, sell and disseminate pornographic materials. MoE: Find trouble. Shared Bottom:Intentionally hurt.	Tags:287 PLE:287 TOP-JUDGE: 287 MMoE:312 MoE:152 Shared Bottom: 204	Tags:11 PLE:11 TOP-JUDGE:11 MMoE:11 MoE:5 Shared Bottom: 11

4. Conclusion

This paper gives a problem description of legal verdict prediction and establishes the model objectives. Based on the encoder-decoder framework, the attention mechanism is utilized to encode the key information of the case, and the joint task of crime prediction and law recommendation is used to construct the multi-task learning model. Bring in the problem of confusing charges to validate the multitask learning model for legal verdict prediction.

The multitask learning legal verdict prediction model proposed in this paper performs well in all three subtasks in LJP.The prediction means of the model in the three subtasks in the CAIL-Small dataset are 85.47%, 80.91%, and 40.88%, respectively. Comparing the experimental results of different keyword encoding methods, the keyword extraction of charges using TF-IDF, TextRank in this paper can solve the problem of confusing charges. And the use of vector summation to combine the offense keywords with the case encoding can effectively distinguish the confusing charges.The experimental F1 value of TF-IDF keyword extraction is 89.13%, and the accuracy rate is 85.68%. The legal judgment prediction model in this paper is used for case fact case prediction, and the multitask learning model incorporating the keyword method has the same prediction results and actual values on the three subtasks.

References

[1] Mack, K., Wallace, A., & Anleu, S. L. R. (2012). Judicial workload: Time, tasks and work organisation. Australasian Institute of Judicial Administration Incorporated.

[2] Dimitrova-Grajzl, V., Grajzl, P., Sustersic, J., & Zajc, K. (2012). Court output, judicial staffing, and the demand for court services: Evidence from Slovenian courts of first instance. International review of law and economics, 32(1), 19-29.

[3] Danziger, S., Levav, J., & Avnaim-Pesso, L. (2011). Extraneous factors in judicial decisions. Proceedings of the National Academy of Sciences, 108(17), 6889-6892

[4] Barry, B. M. (2020). How judges judge: Empirical insights into judicial decision-making. Informa Law from Routledge.

[5] Long, S., Tu, C., Liu, Z., & Sun, M. (2019). Automatic judgment prediction via legal reading comprehension. In Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18 (pp. 558-572). Springer International Publishing.

[6] 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).

[7] Shang, X. (2022). A computational intelligence model for legal prediction and decision support. Computational Intelligence and Neuroscience, 2022(1), 5795189.

[8] Avery, J. (2022). Predicting, up and down: A Framework for Legal Prediction. U. Pa. J. Const. L., 24, 480.

[9] Cui, J., Shen, X., & Wen, S. (2023). A survey on legal judgment prediction: Datasets, metrics, models and challenges. IEEE Access.

[10] Alcántara Francia, O. A., Nunez-del-Prado, M., & Alatrista-Salas, H. (2022). Survey of text mining techniques applied to judicial decisions prediction. Applied Sciences, 12(20), 10200.

[11] Medvedeva, M., Vols, M., & Wieling, M. (2020). Using machine learning to predict decisions of the European Court of Human Rights. Artificial Intelligence and Law, 28(2), 237-266.

[12] Scherer, M. (2019). Artificial Intelligence and Legal Decision-Making: The Wide Open?. Journal of international arbitration, 36(5).

[13] Dong, Q., & Niu, S. (2021, July). Legal judgment prediction via relational learning. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 983-992).

[14] Divya, G., Saxena, N., & Sharma, A. (2024, March). Crime Prediction and Analysis using Machine Learning. In 2024 International Conference on Trends in Quantum Computing and Emerging Business Technologies (pp. 1-5). IEEE.

[15] Taha, K. (2024). Empirical and Experimental Insights into Data Mining Techniques for Crime Prediction: A Comprehensive Survey. ACM Transactions on Intelligent Systems and Technology.

[16] Mussiraliyeva, S., & Baispay, G. (2024). Leveraging Machine Learning Methods for Crime Analysis in Textual Data. International Journal of Advanced Computer Science & Applications, 15(4).

[17] Yuan, D. (2020, October). Case Study of Criminal Law Based on Multi-task Learning. In 2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE) (pp. 98-103). IEEE.

[18] Yao, F., Sun, X., Yu, H., Yang, Y., Zhang, W., & Fu, K. (2020). Gated hierarchical multi-task learning network for judicial decision prediction. Neurocomputing, 411, 313-326.

[19] 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.

[20] Wang, C., & Jin, X. (2020, June). Study on the multi-task model for legal judgment prediction. In 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA) (pp. 309-313). IEEE.

[21] Le, Y., Xiao, S., Xiao, Z., & Li, K. (2024). Topology-aware Multi-task Learning Framework for Civil Case Judgment Prediction. Expert Systems with Applications, 238, 122103.

[22] Xu, Z., Li, X., Li, Y., Wang, Z., Fanxu, Y., & Lai, X. (2020). Multi-task legal judgement prediction combining a subtask of the seriousness of charges. In Chinese Computational Linguistics: 19th China National Conference, CCL 2020, Hainan, China, October 30–November 1, 2020, Proceedings 19 (pp. 415-429).

Springer International Publishing.

[23] Zhang, Y., Wei, X., & Yu, H. (2024). HD-LJP: A Hierarchical Dependency-based Legal Judgment Prediction Framework for multi-task learning. Knowledge-Based Systems, 112033.

[24] Cheng, Y., Chen, C., & Wang, Y. (2024, June). Research on Legal Judgment Prediction with Multi-Task Learning and Causal Logic. In 2024 6th International Conference on Electronic Engineering and Informatics (EEI) (pp. 399-402). IEEE.

[25] Guo, X., Zao, F., Shen, Z., & Zhang, L. (2023). Legal judgment prediction via optimized multi-task learning fusing similarity correlation. Applied Intelligence, 53(21), 26205-26229.

[26] Bi Sheng,Zhou Zhiyao,Pan Lu & Qi Guilin. (2022). Judicial knowledge-enhanced magnitude-aware reasoning for numerical legal judgment prediction. Artificial Intelligence and Law(4),773-806.

[27] Guo Xiaoding,Zao Feifei,Shen Zhuo & Zhang Lei. (2023). Legal judgment prediction via optimized multi-task learning fusing similarity correlation. Applied Intelligence(21),26205-26229.

[28] Wu Geng-Kun,Xu Jie,Zhang Yi-Dan,Wen Bi-Yao & Zhang Bei-Ping. (2024). Weighted feature fusion of dual attention convolutional neural network and transformer encoder module for ocean HABs classification. Expert Systems With Applications122879-.