Legal Cause Prediction with Inner Descriptions and Outer Hierarchies

Zhiyuan Liu[1] [0000−0003−1206−9315], Cunchao Tu[2] [0000−0002−4538−9860],
Zhiyuan Liu[2] [0000−0002−7709−2543], and Maosong Sun[2]

[1] School of Science, Xi'an Jiaotong University, Xi'an, China
[2] Department of CST, Tsinghua University, Beijing, China
{acharkq, tucunchao}@gmail.com
{liuzy, sms}@tsinghua.edu.cn

Abstract. **L**egal **C**ause **P**rediction (LCP) aims to determine the charges in criminal cases or types of disputes in civil cases according to the fact descriptions. The research to date takes LCP as a text classification task and fails to consider the outer hierarchical dependencies and inner text information of causes. However, this information is critical for understanding causes and is expected to benefit LCP. To address this issue, we propose the **H**ierarchical **L**egal **C**ause **P**rediction (HLCP) model to incorporate this crucial information within the seq2seq framework. Specifically, we employ an attention-based seq2seq model to predict the cause path and utilize the inner text information to filter out noisy information in fact descriptions. We conduct experiments on 4 real-world criminal and civil datasets. Experimental results show that our model achieves significant and consistent improvements over all baselines.

1 Introduction

With the release of more than 60 million legal documents from China Judgment Online[3], the analysis, and research on these well-structured and informative legal documents have attracted a wide range of researchers from legal and NLP fields. Among existing works, cause prediction is a representative and fundamental task which aims to predict charges in criminal cases or types of disputes in civil cases. It can provide an effective reference for judges and benefit a series of real-world applications, such as automatic sentencing system and intelligent judgment assistant.

At the early stage, researchers utilize shallow textual features (e.g., characters, words, and phrases) [20, 21] or well-designed features (locations, terms, and dates) [12] to predict charges in criminal cases.

With the successful application of deep learning methods in NLP area [14, 32, 36], researchers propose to employ deep learning techniques to predict causes according to the fact descriptions in legal cases. For example, Luo et al. [23] employ attention mechanism to predict charges with the consideration of relevant

[3] http://wenshu.court.gov.cn/

law articles. Ye et al. [37] utilize seq2seq model for court view generation with additional charge information. Hu et al. [10] introduce several discriminative attributes as internal mappings to predict few-shot and confusing charges.

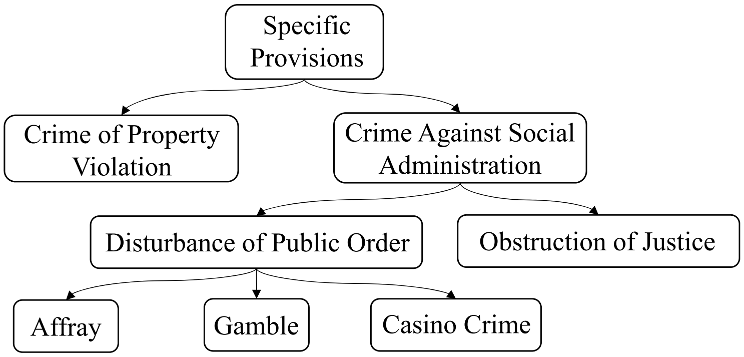

Specific Provisions
Crime of Property Violation
Crime Against Social Administration
Disturbance of Public Order
Obstruction of Justice
Affray
Gamble
Casino Crime

Fig. 1. Hierarchical structure of causes.

It is worth pointing out that all the existing works take cause prediction as a typical text classification task which is usually confronted with two major challenges. First, existing works ignore the implicit relations among causes. As shown in Fig 1, there exists a hierarchical dependency among both criminal and civil causes. The hierarchy provides effective information for all causes, especially for the confusing and few-shot ones. For each leaf cause node, a unique path connects it with the root node. Besides, each cause is treated as a plain symbol by existing classification models and thus the inner information hidden in their names is missed. However, the causes in the legal area are usually well-defined and each of their names can be treated as an accurate and refined description. According to our statistics, the average lengths of criminal and civil cause names are 4.65 and 4.24 words. This critical text information is expected to be used to filter out the noisy information and retain the relevant parts in fact descriptions.

To address these issues, we propose **H**ierarchical **L**egal **C**ause **P**rediction (HLCP) to incorporate the outer hierarchical relations and inner text information of causes. Specifically, we first transform each cause into a path from the root node to leaf cause node. Then, we propose an inner text attention-based seq2seq model to predict cause paths according to the fact descriptions. Compared with traditional hierarchical classification models, HLCP trains only one *global* classifier hierarchically and is able to share knowledge among all nodes in the hierarchy. Experimental results on several large-scale real-world datasets demonstrate that our model outperforms all existing charge prediction models [23, 10] and hierarchical classification models [38, 34, 4].

We publish our code at https://github.com/acharkq/HLCP for further explorations.

2 Related Work

2.1 Cause Prediction

Automatic legal judgment has become a research topic for decades. Kort [17] pioneers the quantitative study on judicial decision prediction by analyzing factual elements numerically. Segal [25] constructs a probabilistic model with variables to evaluate the fairness of the court decisions. These early works [33, 24, 13, 18] usually focus on statistical and mathematical analysis on specific legal scenarios.

Recent works formalize legal judgment as a text classification problem. They focus on scenarios where the judgment result is selected from a fixed label set, e.g. the charges prediction for criminal cases. Machine learning methods are widely utilized as it shows great effectiveness in many areas. These works usually focus on extracting efficient features from facts. Shallow textual features and manually designed patterns are drawn from fact descriptions and annotations (e.g., type, location, term) of cases [21, 19, 1, 28, 22]. Due to the human efforts required for pattern design and annotation, these methods suffer from the issue of scalability.

With the rapid development of deep learning techniques, various neural models [2, 32, 36] show up with a promising performance in NLP tasks. In the legal area, Luo et al. [23] incorporate law article knowledge for cause prediction with an attention-based neural model. Inspired by this work, we utilize the names of causes as the attention queries to filter out noisy information in fact description. Moreover, we introduce the direct hierarchical dependencies among causes into LCP, which could alleviate the data imbalance issue and help to distinguish confusing causes under different parent nodes. Ye et al. [37] employ a seq2seq model in charge prediction task. Different from our work, their seq2seq model is employed for the generation of court views. Hu et al. [10] achieve decent performance promotion on few-shot charge prediction with the annotations of 10 discriminative legal attributes of criminal charges. However, their annotations only cover 32% criminal charges in China. Our method utilizes the off the shelf dependencies among all causes.

2.2 Hierarchical Classification

Hierarchical classification methods have been developed in various application domains. Silla et al. [26] summarizes and classifies these methods into, *flat*, *local*, and *global* ones. *Flat* classifiers ignore the hierarchy structure and treat the problem as a simple multi-class classification. *Local* methods can be classifies into: *local per node*, training binary classifier on each class [8], *local per parent*, training one multi-class classifier for all child nodes of a parent node [27], and *local per level*, training one multi-class classifier at each level instead of each node [7]. These methods enforce the inference to be consistent with the hierarchy, whereas our model is trained under the constraint of the hierarchy. The last method, *global* [16], trains a multi-class classifier which is responsible for the classification of all causes in the hierarchy and employs a loss function that reflects the similarity among labels. In a similar way, we minimize the loss for parent nodes and leaf nodes at the same time and expect this could help to capture the semantic dependencies among causes.

Neural models are also employed for hierarchical classification in NLP tasks. Cerri et al. [5] employs perceptron as classifier for each parent node of the *local per parent node* method. Karn et al. [11] propose to use RNN based encoder-decoder model [6, 30] for entity mention classification task. This model follows the hierarchical path and utilizes attention to filter out noise information level by level. Inspired by the advantage of the seq2seq model for hierarchical dependency modeling, we propose the Hierarchical Legal Cause Prediction model for legal cause prediction.

3 Hierarchical Legal Cause Prediction

In this section, we first give a definition of the LCP task. Next, we introduce the HLCP model.

3.1 Problem Formulation

Table 1. Example of the fact description.

In the early morning of March 28, 2014, the defendant Jia came into Kaiwen restaurant in Shijiazhuang City after drinking. He beat one waiter for no reason. Then, the other waiters of the restaurant started to fight with him...

As shown in Table 1, a fact description refers to the plain description part of a legal document, which is nearly independent of the court's opinion. We regard it as a word sequence $\mathbf{x} = \{x_1, \ldots, x_m, \ldots, x_M\}$, where each word $x_m \in V$. The LCP task aims to predict its corresponding legal cause y, which locates at the bottom of the cause hierarchy.

As shown in Fig. 1, by tracing along the tree-structured hierarchy of causes, we transform the cause label y into a path from root node to cause node, i.e., a label sequence $\mathbf{y} = \{y_1, \ldots, y_i, \ldots, y_I\}$, where $y_i \in Y$. Y denotes the set of all causes in the hierarchy. Note that, each cause y_i in this hierarchy owns its name $s_y = \{x_1, \ldots, x_l, \ldots, x_{L_y}\}$, which can be regarded as a short description of this cause.

With the above denotations, HLCP defines the prediction probability of $\mathbf{y}$ as follows:

$$p(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{I} p(y_i|y_{1:i-1}, \mathbf{x}), \tag{1}$$

As shown in Fig. 2, HLCP consists of two components, i.e., *fact encoder* and *cause predictor*, which will be introduced in the following parts.

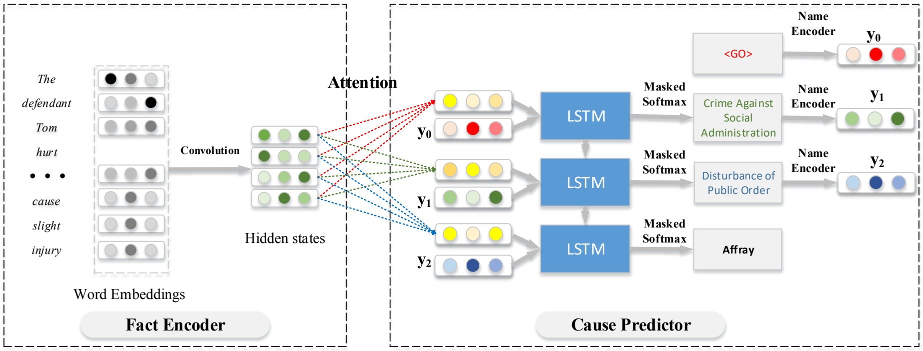


Fig. 2. The framework of HLCP.

3.2 Fact Encoder

As shown in Fig 2, *fact encoder* transforms the word sequence of the fact description into vector representation as the input of *cause predictor*.

Word Representation For a given fact description $\mathbf{x} = \{x_1, \ldots, x_m, \ldots, x_M\}$, we first convert each word x_m to a k-dimensional embedding $\mathbf{x_m}$ by looking up a table $W \in \mathbb{R}^{|V| \times k}$, where $|V|$ denotes the size of the vocabulary. Thus, the input is represented by an embedding sequence:

$$\hat{\mathbf{x}} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m, \ldots, \mathbf{x}_M\}. \tag{2}$$

Convolution Convolutional Neural Network is used after word representation. For the embedding sequence, let $\mathbf{u}_m \in \mathbb{R}^{w \times k}$ denotes the concatenation of w word embeddings $\mathbf{x}_{m:m+w-1}$, where w is the filter width. We then generate $\mathbf{v}_m \in \mathbb{R}^{k_f}$ by

$$\mathbf{v_m} = \mathrm{Relu}(\mathbf{W} \cdot \mathbf{u}_m + \mathbf{b}) \tag{3}$$

where $\mathbf{W} \in \mathbb{R}^{k_f \times (w \times k)}$ and $\mathbf{b} \in \mathbb{R}^{k_f}$. k_f denotes the filter size of CNN. After the convolution step, we obtain $\hat{\mathbf{v}} = \{\mathbf{v}_1, \ldots, \mathbf{v}_m, \ldots, \mathbf{v}_M\}$. This hidden state sequence is used as the attention values of **cause attention**.

Max-Pooling We max pooled over $\hat{\mathbf{v}}$ alone the sequence length axis to obtain the initial state $\mathbf{h}_0$ for *cause predictor* according to the equation:

$$h_{0,j} = \max(\mathbf{v}_{1,j}, \ldots, \mathbf{v}_{M,j}), \forall j \in [1, k_f]. \tag{4}$$

3.3 Cause Predictor

Owing to the successful usage of LSTM [9] on sequence generation, it is adopted for cause sequence prediction in HLCP.

Specifically, the max-pooling output of *fact encoder* is used as the initial state for the LSTM cell in *cause predictor*.

The input of LSTM cell at the i-th step consists of two parts, i.e., **cause representation** and **cause-aware fact representation**. Here, cause representation is the representation for the name of the cause, which is predicted at the previous step. It is calculated by **Name Vectorizer** as follows.

Name Vectorizer As mentioned above, we treat the name of each cause as its short description, which reflects how it differs from other causes. Thus, for the i-th cause y_i, we define **Name Vectorizer** to get its representation $\mathbf{y}_i$ according to its name s_{y_i}, i.e., $\mathbf{y}_i = \mathrm{vec}(s_{y_i})$. In practice, for each name s_{y_i}, we employ an LSTM to encode the corresponding word sequence and take the final hidden state as the cause representation.

Cause Attention Fact description usually contains a large amount of irrelevant and noisy information which could mislead the predictor. The obtained cause representation provides a suitable way to address this issue. Therefore, for each step i, we utilize the cause representation $\mathbf{y}_{i-1}$ at step $i-1$ to select the most relevant information from the fact vectors $\hat{\mathbf{v}}$. In a nutshell, we employ the following cause attention mechanism [3] to get **cause-aware fact representation**.

As illustrated in Fig. 2, we employ Bahdanau Attention to calculate the weight vector α_i of hidden states $\hat{\mathbf{v}}$ as follows:

$$\begin{aligned} e(\mathbf{y}_{i-1}, \mathbf{v}_j) &= \mathbf{U}\tanh(\mathbf{W_0}\mathbf{y}_{i-1} + \mathbf{W_1}\mathbf{v}_j + \mathbf{b}), \\ \alpha_{i,j} &= \frac{\exp(e(\mathbf{y}_{i-1}, \mathbf{v}_j))}{\sum_{t\in[1,M]}\exp(e(\mathbf{y}_{i-1}, \mathbf{v}_t))}. \end{aligned} \tag{5}$$

Here, $\mathbf{U}$, $\mathbf{W_o}$, $\mathbf{W_1}$, and $\mathbf{b}$ are trained parameters. With the obtained weight vector α_i, we can calculate cause-aware fact representation $\mathbf{c}_i$ at step i as follows:

$$\mathbf{c}_i = \sum_{j=1}^{M} \alpha_{ij}\mathbf{v}_j. \tag{6}$$

With the cause representation $\mathbf{y}_{i-1}$ and the cause-aware fact representation $\mathbf{c}_i$ at step i, we concatenate and feed them into LSTM cell to get current hidden state $\mathbf{h}_i$, which will be used for predicting the cause in the current level later.

Masked Classification To ensure that the predicted cause sequence is consistent with the tree structure, we employ masked classification to restrict the prediction scope. That means the causes in the current prediction scope must be the child nodes of the previously predicted cause. For the output $\mathbf{h}_i$ of predictor cell at each step, we employ a weight shared fully connected layer to project it into the legal cause space. Then, the logit of each cause, that is not a child node of the parent cause y_{i-1}, is masked to negative infinity. Thus, the probability of these causes will be 0 after the softmax operation. The above operations can be formalized as:

$$\begin{aligned} p(y_i|y_{1:i-1}, \mathbf{x}) &= \mathrm{softmax}(\mathrm{mask}(\mathbf{W}_p \mathbf{h}_i + \mathbf{b}_p, y_{i-1})), \\ \mathrm{mask}(\mathbf{x}_j, y) &= \begin{cases} \mathbf{x}_j & \pi(Y_j) = y \\ -\infty & \pi(Y_j) \neq y. \end{cases} \end{aligned} \tag{7}$$

where $\mathbf{W}_p$ and $\mathbf{b}_p$ are parameters of the fully connected layer. $\pi(y)$ is the parent cause of y.

3.4 Optimization

For each instance, the training objective of HLCP is to minimize the cross-entropy between predicted cause sequences and the ground-truth as follows:

$$\mathcal{L} = -\frac{1}{I}\sum_{i=1}^{I} \hat{y}_i \log(p(y_i|y_{1:i-1}, \mathbf{x})), \tag{8}$$

where $\hat{y}_i$ is an indicator vector, i.e., $\hat{y}_{ij} = 1$ if $y_i = j$ and $\hat{y}_{ij} = 0$ otherwise.

4 Experiments

To demonstrate the effectiveness of our proposed HLCP model, we conduct experiments on several criminal and civil datasets.

4.1 Datasets

Criminal Case We conduct experiments on the following previous released criminal datasets: CAIL, PKU, and FSC. CAIL (Chinese AI and Law) [35] is a large-scale criminal dataset released for legal competition, we employ the released small version. Zhong et al. [39] collect and construct PKU with criminal cases from Peking University Law Online[4]. FSC (Few-Shot Charge) is a criminal dataset collected by Hu et al. [10] from China Judgment Online[5] for the purpose of few-shot learning. As the cases which contain multiple defendants and multiple causes will increase the difficulty of matching facts to different defendants or causes in LCP task, we filter out these cases in all criminal datasets so that we can focus on the exploration of the validity of introducing hierarchy and text information among causes into LCP task. We set the minimum cause frequency to 30 for CAIL and PKU and 10 for FSC.

[4] http://www.pkulaw.com/

[5] http://wenshu.court.gov.cn/

Civil Case In civil cases, legal cause refers to the type of disputes between parties. Courts will apply proper civil causes to cases based on the fact description and the claim of the plaintiff. Since there are no publicly available civil datasets in previous works for cause prediction, we collect $599,400$ cases from China Judgment Online[5]. We firstly extracted fact descriptions with regular templates from documents automatically. Further, we checked the extraction results of randomly sampled cases and there were few extraction errors. We thereafter filtered out civil cases with multiple causes for the same reason as criminal cases. However, the fact descriptions of civil cases always focus on the interactions among multiple people, instead of recording the action of the defendant in criminal cases. Therefore, we keep civil cases with multiple defendants. The minimum cause frequency is set to 10. The detailed statistics of all datasets are shown in Table 2.

Table 2. The statistics of different datasets.

Datasets	CAIL	PKU	FSC	CIVIL
#Cases	147992	170143	383697	599400
Ave Case Len	225	344	297	390
#Leaf Causes	171	80	149	328
#Parent Causes	26	22	23	95
Min Freq	30	30	10	10

4.2 Baselines

For comparison, we employ 3 typical text classification models: TFIDF+SVM (SVM) [31], TextCNN (CNN) [14], and LSTM [9]; 2 hierarchical classification models: CSSA [4] and Top-Down SVM (TDS) [34]; 2 charge prediction models: Fact-Law Attention (FACT) [23] and Attribute Charge (ATCH) [10] as baselines. We only employ Attribute Charge as baseline on FSC, as it does not cover the cause annotation on other datasets.

4.3 Experimental Settings

As corresponding word embedding is released together with FSC, we conduct experiments followed the experimental settings of Attribute Charge [10] rigorously on FSC. Thus, we can compare with the published performance of Attribute Charge directly.

For other datasets, we employ THULAC [29] for word segmentation. The maximum document length is set to 500 words. Pre-trained 256 dimensional word embedding is employed for word representation. We set the hidden size to 256 for all LSTM cells and set filter widths in CNN to {2,3,4,5} with each filter size to 64. For our model, we set the filter width to 5, filter size to 256, beam width to 5.

For training, we employ Adam [15] as the optimizer and the learning rate is set to 0.001. The batch size is set to 128 and drop out rate is set to 0.5.

Widely used *accuracy* (AC), *macro-precision* (MP), *macro-recall* (MR) and *macro-*F_1 (F_1) as evaluation metrics.

4.4 Results and Analysis

Table 3. Experimental results (- indicates the model can not converge in 150 epochs; * indicates the model is not applicable on the dataset).

Data	CAIL				PKU				CIVIL				FSC			
Metric	AC	MP	MR	F_1	AC	MP	MR	F_1	AC	MP	MR	F_1	AC	MP	MR	F_1
SVM	79.8	76.0	69.7	70.4	95.6	**83.3**	72.5	76.4	85.7	65.6	47.8	52.9	94.4	65.5	54.9	57.7
TDS	77.8	73.8	67.0	68.0	95.1	82.3	69.2	73.1	84.5	63.4	45.9	50.1	93.8	65.4	54.5	57.8
CSSA	63.5	41.2	34.7	38.9	78.5	65.7	62.3	59.4	57.5	42.9	39.1	38.3	83.9	58.8	56.9	54.5
CNN	68.7	62.1	55.5	56.3	96.2	81.4	77.0	76.8	85.4	**67.8**	46.1	52.3	94.5	66.8	58.7	60.7
LSTM	82.5	**79.2**	73.3	73.5	**96.5**	82.4	77.5	78.1	86.3	62.6	49.4	52.9	95.0	68.0	66.7	65.5
FACT	75.9	63.8	61.9	60.4	94.7	76.3	71.4	72.2	-	-	-	-	95.7	73.3	67.1	68.6
ATCH	*	*	*	*	*	*	*	*	*	*	*	*	95.8	75.8	**73.7**	73.1
HLCP	**83.1**	77.4	**74.7**	**74.6**	**96.5**	82.8	**77.8**	**79.3**	**87.3**	66.6	**56.2**	**59.3**	**95.9**	**78.8**	73.5	**74.7**

As shown in Table 3, we compare HLCP with baselines on 4 LCP datasets. We can observe that:

(1) HLCP significantly outperforms the baselines on most metrics and datasets, which proves the robustness and practicability of our model;

(2) Specifically, HLCP improves F_1 by 6% in the CIVIL dataset, which contains the most parent causes. Moreover, HLCP outperforms Attribute Charge which requires human designed charge attributes. The experiments demonstrate the effectiveness of utilizing pre-existing legal cause hierarchy and text knowledge in LCP task;

(3) Conventional hierarchical classification methods perform poorly, while HLCP performs much better with a *global* classifier and the consideration of inner text information;

(4) The improvements on "AC" are limited compared with F_1. The reason is that the distribution of causes is extremely unbalanced. According to our statistics, the most frequent 10 causes cover $17.3\%, 86.9\%, 78.1\%$, and 71.6% cases in CAIL, PKU, FSD, and CIVIL datasets, respectively. While the least common 10 causes only cover $0.22\%, 0.20\%, 0.027\%$, and 0.023% cases. From another perspective, it verifies that HLCP outperforms other methods mainly on the few-shot causes.

By tracing along hierarchy branches, we can transform the output of all models into a cause sequence. Thus, we are enabled to compare the performance

Table 4. Results of causes on different levels.

CAIL	First Level				Second Level				Third level			
Metric	AC	MP	MR	F_1	AC	MP	MR	F_1	AC	MP	MR	F_1
CNN	83.4	78.4	74.9	76.1	75.1	62.9	56.1	57.0	73.3	65.4	60.0	60.6
LSTM	91.4	88.2	**87.6**	87.8	86.1	**78.1**	72.5	72.8	84.1	79.9	**78.9**	77.5
SVM	89.6	83.0	78.9	79.9	83.7	73.2	68.3	68.7	82.3	81.0	74.4	75.3
TDS	88.1	84.9	77.5	78.9	81.9	72.9	65.7	66.9	80.7	77.4	71.7	72.6
HLCP	**91.8**	**89.8**	86.4	**87.9**	**86.6**	76.7	**74.2**	**74.3**	**85.1**	**81.2**	78.1	**77.8**

of HLCP and baselines at different hierarchy level. As shown in Table 4, HLCP defeats baselines in most situations. This affirms our idea of training one *global* classifier for causes in the whole hierarchy benefits the performance on the causes of all levels.

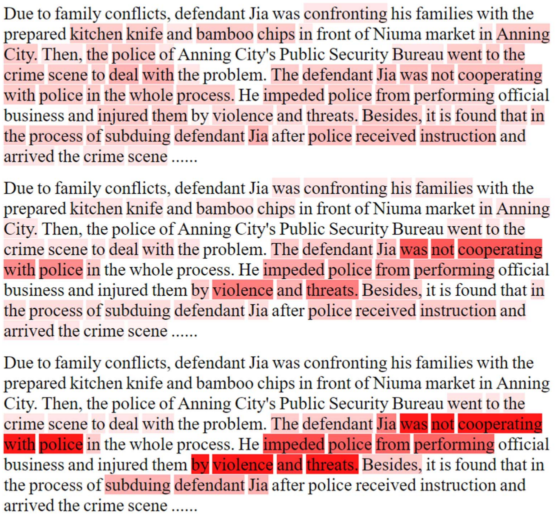
Table 5. Ablation test.

Datasets	CAIL		CIVIL	
Metric	AC	F_1	AC	F_1
w/o name vectorizer	82.7	74.8	87.2	58.5
w/o cause attention	82.6	72.9	**87.5**	59.2
w/o mask	82.1	69.8	86.6	57.0
HLCP	**82.9**	**75.0**	87.3	**59.3**

Ablation Test: To verify the importance of different modules in HLCP, we conduct ablation test as shown in Table 5. Note that, when the name vectorizer is removed, we use randomly initialized vectors as the cause representations. We can observe that all the components in HLCP, including the name vectorizer, cause attention and masked classification, benefit the performance of HLCP. The joint utilization of all modules guarantees the effectiveness of our model.

4.5 Case Study

To give an intuitive illustration of how the name vectorizer and cause attention mechanism works, we select a representative case and visualize the attention results in Fig 3. The fact description records how the defendant conflicted with the public servant. The cause path of this case is "crime against social administration→disturbance of public order→interference with public servant". As shown in Fig. 3, while the cause becomes more and more specific, the attention results turn to be more and more focused. It is consistent with our assumption that irrelevant information, e.g. the word "injured", in fact descriptions is filtered out according to the cause names level by level.



Due to family conflicts, defendant Jia was confronting his families with the prepared kitchen knife and bamboo chips in front of Niuma market in Anning City. Then, the police of Anning City's Public Security Bureau went to the crime scene to deal with the problem. The defendant Jia was not cooperating with police in the whole process. He impeded police from performing official business and injured them by violence and threats. Besides, it is found that in the process of subduing defendant Jia after police received instruction and arrived the crime scene ......

Due to family conflicts, defendant Jia was confronting his families with the prepared kitchen knife and bamboo chips in front of Niuma market in Anning City. Then, the police of Anning City's Public Security Bureau went to the crime scene to deal with the problem. The defendant Jia was not cooperating with police in the whole process. He impeded police from performing official business and injured them by violence and threats. Besides, it is found that in the process of subduing defendant Jia after police received instruction and arrived the crime scene ......

Due to family conflicts, defendant Jia was confronting his families with the prepared kitchen knife and bamboo chips in front of Niuma market in Anning City. Then, the police of Anning City's Public Security Bureau went to the crime scene to deal with the problem. The defendant Jia was not cooperating with police in the whole process. He impeded police from performing official business and injured them by violence and threats. Besides, it is found that in the process of subduing defendant Jia after police received instruction and arrived the crime scene ......

Fig. 3. Attention results of various causes.

Table 6 aims to explain why doing prediction along the hierarchy would be better compared with doing it flat. The target and confusing causes of the listed three cause pairs all belong to different parent causes. However, the causes of each pair still share similarities on fact descriptions. For example, "Gather to disturb social order" belongs to "Disturbance of public order", and "Destruction of production" belongs to "Encroachment of property". But the defendant who breaks the social order usually impacts the production of adjacent businesses. Thus, the causes of this pair are hard to be distinguished from each other. The key difference between them actually refers to the difference between their parent causes. What's more, the classification between parent causes is usually easier. Thus, by doing the prediction hierarchically, we could distinguish confusing causes which belong to different branches at their parent level and avoid the situation in which to make choice between target cause and the confusing one which belongs to another parent cause. As shown in Table 6, HLCP achieves significant performance improvements compared with a flat classifier.

Table 6. Example for confusing causes. (Recall)

Target Cause	Confusing Cause	HLCP	LSTM
Gather to disturb social order	Destruction of production	71.4%	46.4%
Loan fraud	Financial documents fraud	66.7%	53.3%
Kidnap	Racketeering	73.6%	71.7%

4.6 Error Analysis

We summarized the prediction error to 3 reasons:

Data Imbalance This would be the primary reason for failed predictions. The hierarchy we introduced could alleviate this phenomenon by dividing causes into smaller groups so that the sample amount for each cause is competing in a smaller region. However, causes like “the crime of privately carving up state-owned property” and “the dispute of duplicate contracts” which only appear around ten times, are still hard to be learned.

Fuzzy Boundary Some causes, like theft and the crime of forcible seizure, are hard to be distinguished in their nature. The main difference between them is that theft is conducted secretly. However, the secrecy of a crime is sometimes hard to be judged in practical application. There are a few such confusing cause pairs, e.g., (embezzlement, duty encroachment), and (embezzle public money, corruption). As these confusing pairs belong to the same parents, HLCP is unable to distinguish them by predicting cause paths.

Incomplete Information We follow existing LCP works and use fact description as the input. However, when multiple causes are applicable, the court would apply a cause with the consideration of the plaintiff’s claim, which is missed in fact description and cannot always be inferred implicitly. We leave the civil cause prediction task which inputs both the fact description and the plaintiff’s claim as a future work.

5 Conclusion and Future Work

In this work, we propose the HLCP model for LCP task. HLCP builds a novel variation of seq2seq model to capture the dependencies among legal causes. A cause vectorizer is employed to encode legal cause names for noise elimination. Experimental results on four large-scale datasets show that HLCP outperforms conventional text classification models and charge prediction models consistently, which demonstrate the effectiveness and robustness of our model.

In the future, we will explore legal intelligence in the following directions: (1) Legal cause prediction with multiple causes and defendants; (2) Incorporating a plaintiff’s claim into civil cause prediction task; (3) Incorporate the logic rules defined by law articles into legal judgment prediction task.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (No. 2018YFC0831900).

References

1. Aletras, N., Tsarapatsanis, D., Preotiuc-Pietro, D., Lampos, V.: Predicting judicial decisions of the european court of human rights: A natural language processing perspective. PeerJ Computer Science **2** (2016)
2. Baharudin, B., Lee, L.H., Khan, K.: A review of machine learning algorithms for text-documents classification. JAIT **1**(1), 4–20 (2010)
3. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: Proceedings of ICLR (2015)
4. Bi, W., Kwok, J.T.: Multi-label classification on tree-and dag-structured hierarchies. In: Proceedings of ICML. pp. 17–24 (2011)
5. Cerri, R., Barros, R.C., De Carvalho, A.C.: Hierarchical multi-label classification using local neural networks. Computer and System Sciences **80**(1), 39–56 (2014)
6. Cho, K., Merrienboer, B.V., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. Computer Science (2014)
7. Clare, A., King, R.D.: Predicting gene function in saccharomyces cerevisiae. Bioinformatics **19**(suppl_2), ii42–ii49 (2003)
8. Fagni, T., Sebastiani, F.: On the selection of negative examples for hierarchical text categorization. In: Proceedings of LTC. pp. 24–28 (2007)
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
10. Hu, Z., Li, X., Tu, C., Liu, Z., Sun, M.: Few-shot charge prediction with discriminative legal attributes. In: Proceedings of COLING (2018)
11. Karn, S., Waltinger, U., Schütze, H.: End-to-end trainable attentive decoder for hierarchical entity classification. In: Proceedings of EACL. vol. 2, pp. 752–758 (2017)
12. Katz, D.M., Bommarito II, M.J., Blackman, J.: A general approach for predicting the behavior of the supreme court of the united states. PloS one **12**(4) (2017)
13. Keown, R.: Mathematical models for legal prediction. Computer/LJ **2**, 829 (1980)
14. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of EMNLP (2014)
15. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of ICLR (2015)
16. Kiritchenko, S., Matwin, S., Nock, R., Famili, A.F.: Learning and evaluation in the presence of class hierarchies: Application to text categorization. In: Proceedings of CSCSI. pp. 395–406 (2006)
17. Kort, F.: Predicting supreme court decisions mathematically: A quantitative analysis of the ”right to counsel” cases. American Political Science Review **51**(1), 1–12 (1957)
18. Lauderdale, B.E., Clark, T.S.: The supreme court's many median justices. American Political Science Review **106**(4), 847–866 (2012)
19. Lin, W.C., Kuo, T.T., Chang, T.J., Yen, C.A., Chen, C.J., Lin, S.d.: Exploiting machine learning models for chinese legal documents labeling, case classification, and sentencing prediction. In: Processdings of ROCLING. p. 140 (2012)
20. Liu, C.L., Chang, C.T., Ho, J.H.: Case instance generation and refinement for case-based criminal summary judgments in chinese. JISE (2004)
21. Liu, C.L., Hsieh, C.D.: Exploring phrase-based classification of judicial documents for criminal charges in chinese. In: Proceedings of ISMIS. pp. 681–690 (2006)

22. Liu, Y.H., Chen, Y.L.: A two-phase sentiment analysis approach for judgement prediction. Journal of Information Science (2017)
23. Luo, B., Feng, Y., Xu, J., Zhang, X., Zhao, D.: Learning to predict charges for criminal cases with legal basis. In: Proceedings of EMNLP (2017)
24. Nagel, S.S.: Applying correlation analysis to case prediction. Tex. L. Rev. **42**, 1006 (1963)
25. Segal, J.A.: Predicting supreme court cases probabilistically: The search and seizure cases, 1962-1981. American Political Science Review **78**(4), 891–900 (1984)
26. Silla, C.N., Freitas, A.A.: A survey of hierarchical classification across different application domains. Data Mining and Knowledge Discovery **22**(1-2), 31–72 (2011)
27. Silla Jr, C.N., Freitas, A.A., et al.: Novel top-down approaches for hierarchical classification and their application to automatic music genre classification. In: SMC. pp. 3499–3504 (2009)
28. Sulea, O.M., Zampieri, M., Vela, M., Genabith, J.V.: Exploring the use of text classification in the legal domain. In: Proceedings of ASAIL workshop (2017)
29. Sun, M., Chen, X., Zhang, K., Guo, Z., Liu, Z.: Thulac: An efficient lexical analyzer for chinese. (2016)
30. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in neural information processing systems. pp. 3104–3112 (2014)
31. Suykens, J.A., Vandewalle, J.: Least squares support vector machine classifiers. Neural processing letters **9**(3), 293–300 (1999)
32. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: Proceedings of EMNLP. pp. 1422–1432 (2015)
33. Ulmer, S.S.: Quantitative analysis of judicial processes: Some practical and theoretical applications. Law & Contemp. Probs. **28**, 164 (1963)
34. Vateekul, P., Kubat, M., Sarinnapakorn, K.: Top-down optimized svms for hierarchical multi-label classification: a case study in gene function prediction. Intelligent Data Analysis (2013)
35. Xiao, C., Zhong, H., Guo, Z., Tu, C., Liu, Z., Sun, M., Feng, Y., Han, X., Hu, Z., Wang, H., et al.: Cail2018: A large-scale legal dataset for judgment prediction. arXiv preprint arXiv:1807.02478 (2018)
36. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A.J., Hovy, E.H.: Hierarchical attention networks for document classification. In: Proceedings of NAACL. pp. 1480–1489 (2016)
37. Ye, H., Jiang, X., Luo, Z., Chao, W.: Interpretable charge predictions for criminal cases: Learning to generate court views from fact descriptions. In: Proceedings of NAACL (2018)
38. Zeng, X., Yang, C., Tu, C., Liu, Z., Sun, M.: Chinese liwc lexicon expansion via hierarchical classification of word embeddings with sememe attention. In: Proceedings of AAAI (2018)
39. Zhong, H., Guo, Z., Tu, C., Xiao, C., Liu, Z., Sun, M.: Legal judgment prediction via topological learning. In: Proceedings of EMNLP (2018)