An Element is Worth a Thousand Words: Enhancing Legal Case Retrieval by Incorporating Legal Elements

Anonymous ACL submission

Abstract

Legal case retrieval plays an important role in promoting judicial justice and fairness. One of its greatest challenges is that the definition of relevance goes far beyond the common semantic relevance as in ad-hoc retrieval. In this paper, we reveal that the legal elements, which typically comprise key facts in a specialized legal context, can largely improve the relevance matching of legal case retrieval. To facilitate the use of legal elements, we construct a Chinese legal element dataset called *LeCaRD-Elem* based on the widely-used LeCaRD dataset (Ma et al., 2021), through a two-stage semi-automatic method with a minimized reliance on human labor.

Meanwhile, we introduce two new models to enhance legal search using legal elements. The first, *Elem4LCR-E*, is a two-stage model that explicitly predicts legal elements from texts and then leverages them for improved ranking. Recognizing the potential benefits of more seamless integration, we further propose an end-to-end model called *Elem4LCR-I*, which internalizes the legal element knowledge into its model parameters using a tailored teacher-student training framework. Extensive experiments underscore the significant value of legal elements and demonstrate the superiority of our two proposed models in enhancing legal search over existing methods. The anonymous code and the sampled dataset are accessible here.

1 Introduction

Legal case retrieval is a crucial part of legal AI, aiming to find the most relevant cases in a case collection for a given query case. In recent years, legal case retrieval has gained increased attention due to the growing significance of the "similar cases, similar decisions" principle in various countries (Hamann, 2019; Bench-Capon et al., 2012).

In legal case retrieval, both the input query and the candidate document are lengthy legal cases.

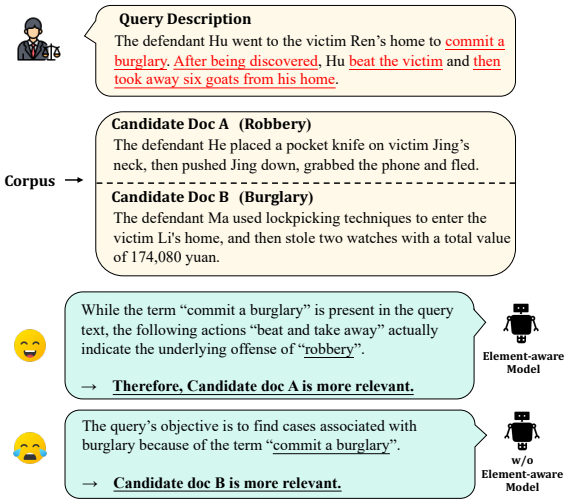

Figure 1: An example of the legal search system. The red lines are key behaviors and facts. Legal elements can assist the model in understanding important legal concepts, enabling it to identify cases that are really relevant in a legal context.

Relevance is assessed based on the similarity between the query and the document case in the specialized and complex legal context. Above semantic similarity, the measurement of relevance in legal case retrieval is usually intricate. This stands in contrast to conventional ad-hoc searches, where queries are often concise, and relevant documents simply address the information needs conveyed in those queries. This disparity introduces substantial challenges when adapting ad-hoc search models for legal case retrieval.

In this paper, we argue that modeling legal elements is critical to legal case retrieval, because (1) Legal elements are specific components to establish guilt, which typically comprise key facts in a specialized legal context. In practice, legal experts may also distill the criminal process into these distinct legal elements based on legal theory (Zhong et al., 2020) and then use them for judicial judgments and supporting case identification. (2) Le-

gal elements can help the model precisely identify pertinent within confusing fact descriptions. As illustrated in Figure 1, the user submits a query regarding the crime of robbery, with the expectation that the system will provide similar cases. For a non-element-aware model, the description "commit a burglary" in the query text is likely to mislead it to rank cases with burglary crimes at the top. But for an element-aware model, the combination of "commit a burglary" and the following description "beat the victim" actually indicates the crime of robbery, leading the cases related to robbery to get high ranking scores.

However, it is non-trivial to obtain high-quality legal element annotation data for large-scale legal cases. Different from laws or crimes which can be clearly stated in authoritative documents, legal elements often have subtle differences across legal theories. To facilitate the study of legal elements, in this work, we propose a semi-automatic annotation approach for legal elements, which is much more effective, efficient, and economical than existing annotation approaches that either completely rely on human experts (Shu et al., 2019) or heuristically rules (Lyu et al., 2022; Zhao et al., 2022). Based on our proposed annotation approach, we annotate the legal elements of a widely-recognized legal case retrieval dataset, i.e., LeCaRD (Ma et al., 2021), and finally contribute a new dataset, called **LeCaRD-Elem** to the community.

Furthermore, we embark on an exploration into the effective utilization of legal elements for legal case retrieval. We present two legal element-aware ranking models, including a two-step **Elem4LCR-E** model and an end-to-end **ELem4LCR-I** model. Specifically, Elem4LCR-E first explicitly predicts legal elements from texts and then concatenates these elements with the original legal case text for ranking. In contrast, ELem4LCR-I can implicitly leverage the legal elements for ranking by learning with a novel multi-level knowledge distillation method under tailored curriculums. The legal element knowledge has been internalized into ELem4LCR-I during training, and thus it does not need to explicitly extract the legal elements of the query and document cases in the inference stage.

We conduct extensive experiments on our proposed LeCaRD-Elem dataset. The experimental results highlight the significant value of legal elements and demonstrate the superiority of the proposed two models (i.e., Elem4LCR-E and Elem4LCR-I) in enhancing legal search with legal elements over existing methods.

In summary, our main contributions are:

(1) We empirically demonstrate that legal elements possess substantial value and potential in improving legal case retrieval.

(2) We propose a more efficient and economical two-stage method for the annotation of legal elements and introduce a well-curated Chinese legal element dataset (LeCaRD-Elem) that can facilitate various downstream legal intelligence tasks.

(3) We pioneer the study of leveraging legal elements for legal case retrieval by proposing an explicit-style model (Elem4LCR-E) and an implicit-style model (Elem4LCR-I). By integrating the legal element information, the models can focus on the more critical information within case descriptions to achieve better matching performance.

2 Related Work

2.1 Legal Case Retrieval

We review a few important legal case retrieval datasets and methods in this section. In terms of datasets, cited-based methods (Kano et al., 2018) construct relevance labels based on supportive cases in query documents. Expert-based methods (Xiao et al., 2018; Ma et al., 2021; Bhattacharya et al., 2019; Locke and Zuccon, 2018) rely on human labor to identify similar cases and try to ensure consistency by setting pre-defined criteria. In terms of legal search methods, classic text retrieval models can be naturally applied to the legal domain and some still serve as strong baseline models (Rosa et al., 2021). Researchers also incorporate additional information and knowledge in the legal domain to enhance search quality (Tran et al., 2020; Saravanan et al., 2009). In recent years, many approaches based on pre-trained language models have made great progress (Chalkidis et al., 2020; Zhong et al., 2019). Additionally, paragraph-level interaction modeling (Shao et al., 2020) and longformer-based pre-training in the legal domain (Xiao et al., 2021) are proposed to handle lengthy legal texts.

2.2 Exploration of Legal Element in Legal AI

In legal theory (Fletcher, 2001), the concept "legal element" refers to specific components to establish guilt, which typically comprise key facts in a specialized legal context. Existing research on legal elements can be divided into two categories: (1) Entirely relying on human expert for annota-

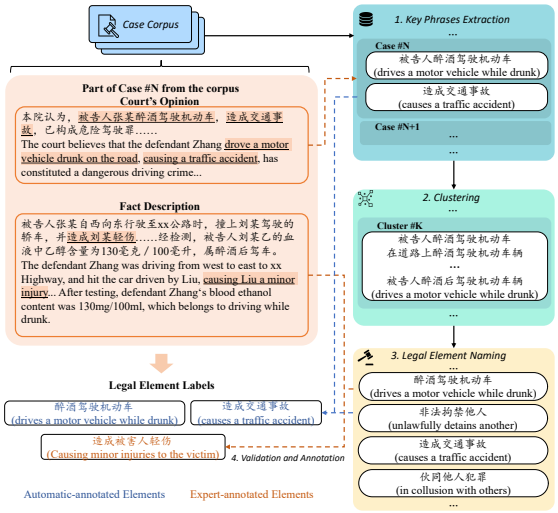

Figure 2: Overview of LeCaRD-Elem construction.

tion: Shu et al. (2019) construct CAIL2019-FE, selecting the cases of divorce dispute, labor dispute, and loan dispute for expert annotation. (2) Rule-based construction from other legal information (e.g., crime (Lyu et al., 2022; Zhao et al., 2022)). However, the crimes of a case often fail to comprehensively encompass all the legal elements within the case. Besides, it is worth noting that legal elements are not directly stated in official documents, and their interpretation may vary among different legal theories. There exist substantial and diverse legal elements and it will be unaffordable or inaccurate to completely rely on experts or heuristic rules to discern them.

3 Legal Element Annotation

In this section, we introduce our two-stage semi-automic approach for curating our LeCaRD-Elem dataset, including a *Data Mining Stage* and an *Expert Annotation stage*. The overview of the data curation process is illustrated in Figure 2.

3.1 Data Mining Stage

The goal of the data mining stage is to obtain a set of clusters from the raw case corpus. Each cluster contains some semantically similar key phrases extracted from the fact descriptions. Every cluster will be a candidate pool for the generation of legal elements as we will illustrate in the next *Expert Annotation* stage.

3.1.1 Key Phrases Extraction

We use the same raw case corpus as LeCaRD (Ma et al., 2021). To begin, we extract key phrases that can effectively characterize legal elements from the raw case corpus through the following four steps: (1) We use Chinese punctuation marks (including commas and periods) to split the court's opinion part into several text snippets. (2) For each snippet, we remove those procedural descriptions (e.g., "the court held that") from it since they lack specific information about the legal case. (3) For each snippet, we remove the person and place names[1] from it to make its content more generalized. (4) We discard the snippets which contain descriptions like "constitute __ crime" because it indicates that this snippet is about the result of judgment rather than legal elements. Finally, after deduplication, each remaining snippet is considered as a *key phrase*.

3.1.2 Clustering

Then, we try to group key phrases which are semantically similar into the same cluster. Specifically, we feed each key phrase into BERT (Devlin et al., 2018) and use the output [CLS] embedding as its semantic representation. Considering that the number of legal elements is unknown, we employ an agglomerative clustering algorithm, Ward (Ward Jr, 1963), to merge similar phrases from bottom to top. The merging procedure stops when the merging distance falls below a pre-defined threshold. We find that the number of clusters tends to converge within a range (approximately 500 clusters in our experiment) as the quantity of case data increases. In practice, due to the standardized nature of judicial statements made by judges in legal cases, our clustering method generally demonstrates good effectiveness. Also, although we focus on Chinese legal cases in this work, our data mining stage can be easily extended to legal systems in other languages that have similar highly standardized statements for legal cases as Chinese, such as German.

3.2 Expert Annotation Stage

3.2.1 Legal Element Naming

For each cluster, we randomly sample dozens of key phrases from it and employ human experts to summarize these sampled key phrases, and finally write one legal element for this cluster. Clusters that cannot be summarized into legal elements (e.g., conveying the semantics "in defiance of the law") are discarded. In our practice, the vast majority of clusters contain at most one legal element. In rare exceptions, with the help of legal experts, we add multiple legal element names corresponding to

[1]The person and place names are automatically detected using Lexical Analyzer for Chinese tool.

Table 1: Statistics of LeCaRD-Elem and CAIL2019-FE.

Dataset	CAIL2019-FE	LeCaRD-Elem
# Documents	2,740	9,195
# Elements	60	475
# Avg. Elements / Case	5.62	4.55
Case Type	Civil	Criminal
Language	Chinese	Chinese

the cluster to the label table and let the annotators choose the real mapped element for this case in the following Section 3.2.2.

3.2.2 Verification and Annotation

Then, the initial legal elements for each legal case are automatically labeled with the legal elements corresponding to the key phrases contained in that legal case. Note that there is a one-to-one relationship between legal elements (or clusters) and key phrases. After the initial automatic legal element annotation, we finally employ a new group of annotators to check and correct the annotation to further improve the annotation quality. Annotators can supplement new elements or remove incorrect elements annotation for each legal case. All annotators have degrees in law. Particularly, we engage legal experts who hold a Ph.D. in law to categorize all legal elements based on legal theory prior to the annotation process, which can help annotators quickly familiarize themselves with these elements.

3.3 Dataset Statistics

The statistics of our curated LeCaRD-Elem dataset are presented alongside a widely used civil legal element dataset, CAIL2019-FE (Shu et al., 2019), in Table 1. Notably, our LeCaRD-Elem dataset comprises 3.36 times more legal cases and 8.38 times more legal elements compared to CAIL2019-FE. While CAIL2019-FE is centered on civil law, our LeCaRD-Elem dataset uniquely addresses the absence of criminal law legal elements. Our LeCaRD-Elem dataset comprises a wide variety of legal elements, and also exhibits a long-tail distribution phenomenon (Hayes and Weinstein, 1990; Tsatsaronis et al., 2015; Coordinators, 2016). We further study the long-tail distribution and typical cases of high and low-frequency elements in LeCaRD-Elem in Appendix A for the page limit.

4 Element-Aware Legal Case Retrieval

In this section, we present two approaches for enhancing legal case retrieval by integrating the knowledge of legal elements.

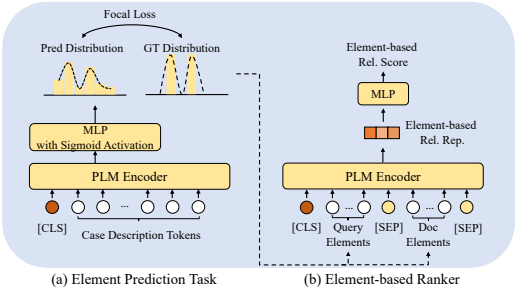

Figure 3: Overview of Elem4LCR-E. The input of the element-based ranker is either the element prediction result or the ground-truth element labels.

4.1 Problem Definition

In this paper, we focus on the re-ranking task of legal case retrieval. Given a query q and a list of candidate documents $D = \{d_1, ..., d_n\}$ recalled from previous stages, our goal is to score each candidate document based on its relevance to the query. The query q is a legal case containing only fact descriptions. Each candidate document is a real legal case whose trial has been completed. In the training stage, each query and candidate document has its own ground-truth legal element labels. However, in the inference stage, element labels will not be provided, aligning with the real scenario.

4.2 Explicit Approach: Elem4LCR-E

We present Elem4LCR-E as a two-step pipeline approach: extracting legal elements from legal texts and then employing a well-trained ranking model.

As shown in Figure 3(a), in the first step, we employ a pre-trained language model BERT to perform multi-legal-element classification based on the case description. The embedding output of the [CLS] token is mapped to $\mathbb{R}^{|\mathcal{E}|}$ using a multi-layer perceptron with the sigmoid function in the final layer, where $\mathcal{E}$ is the set of total legal elements. Then, the legal elements whose prediction probabilities are higher than a pre-defined threshold τ will be retained. After obtaining the predicted legal elements, in the second step, we adopt the cross-encoder architecture (Qiao et al., 2019) to rank the documents. As shown in Figure 3(b), for a query (or a document), we concatenate all of its predicted legal elements into a text sequence. Then, the legal element sequence of the query and the document are concatenated and input into the BERT encoder to get the final relevance score.

The two BERT models are trained using the classical focal loss function (Lin et al., 2017) and the

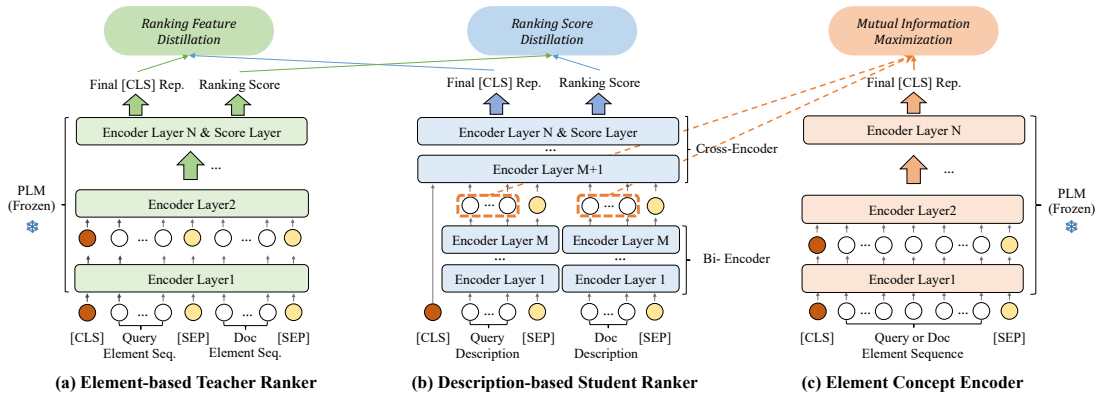

Figure 4: The training framework of Elem4LCR-I. We propose an element knowledge enhancement method based on mutual information maximization, a multi-level element interaction distillation method, and a customized training curriculum to comprehensively improve the model training with legal elements.

pairwise ranking loss function (Nogueira and Cho, 2019), respectively. Detailed training formulas are shown in Appendix B.

4.3 Implicit Approach: Elem4LCR-I

We further introduce a more advanced end-to-end approach Elem4LCR-I, which avoids information loss caused by the explicit element prediction of the first step of Elem4LCR-E.

Specifically, Figure 4 (b) shows the model architecture of Elem4LCR-I. It stacks M bi-encoder layers and $N - M$ cross-encoder layers. The motivation for such a decomposed model architecture is to enhance the understanding of legal element concepts in the lower M layer while learning improved relevance interaction in the higher $(N - M)$ layers. Given a query case q and a document case d, the final ranking score r_{rank} is obtained through:

$$r_{\text{rank}} = \phi(\text{Pool}_{[\text{CLS}]}(\text{CE}(e_{[\text{CLS}]} \circ \text{BE}(q) \circ \text{BE}(d))), \quad (1)$$

where $e_{[\text{CLS}]}$ is the word embedding of the [CLS] token, BE (Bi-encoder) and CE (Cross-Encoder) are the lower M layers and the higher $(N - M)$ layers of the text encoder, respectively. $\text{Pool}_{[\text{CLS}]}$ refers to pooling with the embedding output of the [CLS] token, and $\phi(\cdot)$ is a multi-layer perception.

To facilitate the training of this decomposed legal element-aware model, we design a novel training framework that contains three important aspects: (1) *Element Knowledge Enhancement*, (2) *Multi-level Element Interaction Distillation*, and (3) *Tailored Curriculum Learning Strategy*.

4.3.1 Element Knowledge Enhancement

In our model design, the token representations output from the first M bi-encoder layers can be interpreted as an implicit form of legal element knowledge. To enhance the model's grasp of the legal elements, we propose a method based on mutual information maximization.

Specifically, as shown in Figure 4 (c), given a legal element text sequence t of a query/document case, we first use a frozen BERT encoder to obtain its representation e of its [CLS] token. Suppose that the output token representations of the M-th layer of our model for this query/document are $H = \{h_1, ..., h_l\}$, where l is the query/document token length, we try to maximize the mutual information between each token representation h_i and the legal element representation e using JS-divergence:

$$\mathcal{L}_{\text{MI}} = -\frac{1}{l}\sum_{i=1}^{l}\{\mathbb{E}_{\mathbb{P}(e,h_i)}[-\text{softplus}(-T_\theta(e,h_i))] - \mathbb{E}_{\mathbb{P}(e)\mathbb{P}(h_i)}[-\text{softplus}(-T_\theta(e,h_i))]\}, \quad (2)$$

where $\mathbb{P}(e, h_i)$ indicates the distribution that e and h_i are derived from the same query/document, whereas $\mathbb{P}(e)\mathbb{P}(h_i)$ implies that they are derived from different queries/documents. T_θ is an approximator implemented with a fully connected network. $\text{softplus}(x) = log(1 + e^x)$.

4.3.2 Multi-level Element Interaction Distillation

We adopt the teacher-student paradigm to train our end-to-end model to learn from the legal element interactions towards better ranking performance. As shown in Figure 4 (a)(b), we use the well-trained element-based ranker of Elem4LCR-E, whose input is the concatenation of the ground-truth legal elements of the query and the document, as the teacher ranker. We distill its knowledge of both

element interaction features and ranking prediction logits into the student ranker, i.e., Elem4LCR-I.

Specifically, suppose a training quadruple (q, d, t_q, t_d), where q, d, t_q, and t_d are the query case, document case, the ground-truth legal elements text sequence of the query and the document, respectively. For the feature-level distillation, we use the SmoothL1 loss function to minimize the distance between the [CLS] token representations of the teacher ranker $\mathcal{T}$ and the student ranker $\mathcal{S}$:

$$\mathcal{L}_{\text{feat}} = \text{SmoothL1}\left(\text{Pool}_{[\text{CLS}]}\left(\mathcal{T}(t_q \circ t_d)\right), \text{Pool}_{[\text{CLS}]}\left(\mathcal{S}(q \circ d)\right)\right). \quad (3)$$

For the logit-level distillation, we consider the pairwise training, where we have one positive document $d+$ and Z negative document $\{d_1^-, ..., d_Z^-\}$ for a query q. We distill the normalized logit distribution of the teacher ranker into the student ranker using KL-divergence as the loss function:

$$P^t = \text{softmax}([r^{t^+}, r_1^{t^-}, ..., r_Z^{t^-}]), \quad (4)$$

$$P^s = \text{softmax}([r^{s^+}, r_1^{s^-}, ..., r_Z^{s^-}]), \quad (5)$$

$$\mathcal{L}_{\text{logits}} = D_{\text{KL}}(P^t || P^s), \quad (6)$$

where r^t and r^s are the prediction logits of the teacher ranker and the student ranker, respectively.

4.3.3 Tailored Curriculum Learning

There are a few challenging samples in real-world data that cannot be distinguished based solely on legal elements. Early exposure to these samples during training may result in overfitting. To mitigate this issue, we suggest arranging the training samples from easy to hard in a tailored curriculum for more stable training. Specifically, we propose a rule-based strategy and a model-based strategy to define the sample difficulty. (1) For the rule-based strategy, if the training dataset includes multi-level relevance labels as opposed to binary labels, we consider the negative samples with higher relevance labels as more difficult. (2) For the model-based strategy, we calculate the ratio of logits produced by the teacher model on positive and negative samples, considering those samples with a ratio below a pre-defined threshold τ as hard negatives.

Elem4LCR-I is finally trained with multi-task learning of three objectives under an easy-to-hard curriculum:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{logits}} + \lambda_2 \mathcal{L}_{\text{feat}} + \lambda_3 \mathcal{L}_{\text{MI}}, \quad (7)$$

where λ_1, λ_2, and λ_3 are hyper-parameters to balance the losses.

5 Experiments

5.1 Experimental Settings

Datasets. We conduct experiments on our curated LeCaRD-Elem dataset, which maintains the same queries, documents, and relevance labels with the original LeCaRD (Ma et al., 2021) dataset, but has additional annotations of legal elements for all the query and document cases. Specifically, LeCaRD contains 107 query cases and 43,823 candidate document cases. Document cases of each query have relevance labels ranging from 0 to 3. A document case is considered relevant if its relevance label is 3, otherwise it is considered irrelevant. To alleviate the instability caused by the small number of test samples, we follow the previous work (Yao et al., 2022) to adopt 5-fold cross-validation for evaluation.

Evaluation metrics. We adopt mean average precision (MAP), Precision(P@k, k $\in \{5\}$), normalized discounted cumulative gain (NDCG@k, k $\in \{5, 20, 30\}$) to comprehensively evaluate the ranking performance.

Baselines. We select three types of baseline models: **(1) Traditional ranking methods: BM25** (Robertson et al., 1995), **TF-IDF** (Salton and Buckley, 1988), and **LMIR** (Ponte and Croft, 2017). **(2) Generic neural ranking models based on PLMs: BERT** (Devlin et al., 2018) is pre-trained on a large-scale corpus. The texts are concatenated and then inputted to the model. **NEZHA** (Wei et al., 2019) adopts relative positional encoding and whole word masking techniques based on BERT. **BERT-xs** (Zhong et al., 2019) is pretrained in large-scale Chinese criminal case documents. **(3) Neural ranking models designed for long text problems in legal domain: BERT-PLI** (Shao et al., 2020) uses BERT model to capture the semantic relevance at the paragraph level, and then aggregate local matching signals to obtain relevance scores. **Lawformer** (Xiao et al., 2021) adopts Longformer's model architecture on legal corpus for pre-training in legal texts. We further introduce the implementation details in Appendix D due to the page limit.

5.2 Main Results

The main results are shown in Table 2. Specifically, Elem4LCR-E* denotes Elem4LCR-E when fed with the ground-truth element labels. Since

Table 2: Experimental results on LeCaRD. "†" indicates the model outperforms all baselines significantly with paired t-test at $p < 0.05$ level. The best results are in bold. Particularly, Elem4LCR-E* denotes Elem4LCR-E when fed with the ground-truth element label from LeCaRD-Elem.

Model	MAP	P@5	NDCG@5	NDCG@20	NDCG@30
Traditional ranking baselines					
BM25	47.5	39.6	45.2	55.9	65.3
TF-IDF	44.7	30.3	36.5	40.1	42.6
LMIR	48.8	42.8	46.6	57.3	65.9
General PLM-based neural ranking baselines					
BERT	50.6	45.8	49.9	58.2	68.4
NEZHA	49.8	46.4	48.5	58.2	67.3
BERT-xs	50.5	45.2	50.0	59.4	66.1
Neural ranking baselines designed for long text					
BERT-PLI	51.0	45.0	51.9	59.4	65.1
Lawformer	51.1	45.6	50.4	59.1	64.9
Our methods					
Elem4LCR-E	53.5	43.9	49.8	63.0	70.6
Elem4LCR-I	**55.1**†	**48.2**†	**54.3**†	**63.8**†	**72.1**†
Elem4LCR-E* (upper bound)	63.4	54.8	63.6	73.1	77.9

this setup is inconsistent with real scenarios, we present it merely as an indicative "upper-bound" for Elem4LCR-E for reference. From the results, we can obtain the following observations:

(1) Both Elem4LCR-E and Elem4LCR-I outperform all baselines. This demonstrates the effectiveness of incorporating legal elements for case retrieval. Whether through explicit or implicit methods, legal elements assist the ranker in more accurately identifying relevant cases. We posit that this is because legal elements represent a more essential feature compared to complex fact descriptions.

(2) Elem4LCR-I shows better performance compared to Elem4LCR-E. In our experiments, the precision, recall, and F1 scores of the element prediction task in Elem4LCR-E are 0.652, 0.671, and 0.636, respectively. Elem4LCR-I achieves better performance by preventing information loss in the pipeline. However, it is worth noting that Elem4LCR-E exhibits a substantial discrepancy from its oracle results, indicating its potential for improvement. We further discuss the respective advantages of the two proposed methods in Appendix E.

(3) Neural ranking baselines designed for long text do not show obvious advantages. BERT-PLI and Lawformer are models designed for tackling long text problems. Although these two methods are input with longer texts, they only exhibit limited advantages when compared with BERT. Elem4LCR-I outperforms these two baselines significantly by leveraging legal element knowledge within a limited text length. This demonstrates that existing long-text modeling methods fail to effectively extract relevance signals in lengthy inputs.

Table 3: Ablation study results of Elem4LCR-I.

Ablation	MAP	NDCG@5	NDCG@30
w/o LD	54.3	52.9	71.4
w/o FD	53.2	52.4	70.6
w/o MIM	53.7	52.9	70.7
w/o CL	51.1	49.4	69.3
Only-CL	50.9	49.0	68.7
Elem4LCR-I w/ CL-M	54.8	53.7	71.7
Elem4LCR-I w/ CL-R	**55.1**	**54.3**	**72.1**

(4) Traditional methods (e.g. BM25) are still strong baselines. Although all neural ranking methods outperform traditional methods, traditional methods do not perform badly, which is consistent with the conclusion of previous works (Rosa et al., 2021; Ma et al., 2021). We believe that it is because traditional methods are less affected by document length and complex facts compared to neural ranking methods, thus narrowing the gap between them.

5.3 Ablation Study

Since Elem4LCR-E is a pipeline-style approach, we focus on performing ablation experiments to verify the necessity of each component in Elem4LCR-I. Specifically, we investigate the effectiveness of four components: mutual information maximization (MIM), feature-level distillation (FD), logits-level distillation (LD), and our curriculum learning (CL). For curriculum learning, the model with only curriculum learning strategy (Only-CL), the model-based strategy (CL-M), and

the rule-based strategy (CL-R) are evaluated.

As shown in Table 3, removing any of the existing components leads to a decrease in performance. Interestingly, the removal of CL or Only-CL brings the most obvious performance decrease. This demonstrates that the curriculum learning strategy itself doesn't lead to better performance. Its primary role in Elem4LCR-I is to enhance the model's comprehension of legal elements by providing a more reasonable learning order of samples. The removal of the other three components also results in a performance decrease to varying degrees, which shows that learning element concepts and element-based relevance estimation simultaneously leads to better performance. Besides, the model-based curriculum learning strategy shows comparable performance to the rule-based strategy, indicating that our proposed curriculum is still effective without human annotation.

5.4 Effect of Layer Number of Bi-Encoders

We fix the total number of parameters of our Elem4LCR-I and investigate the effects of using different layer numbers for bi-encoders (i.e., M). Correspondingly, an increase in bi-encoder layers will result in a decrease in the cross-encoder layers. Our goal is to explore the best segmentation locations for these two types of layers.

As shown in Figure 5, we find that the model's performance exhibits a general trend of initially increasing and subsequently decreasing as the number of bi-encoder layers increases. When the number of bi-encoder layers is insufficient, the network capacity is not enough to facilitate the comprehensive capture of legal element concepts. However, excessive layers in the bi-encoder impair the capacity of the cross-encoders, leading to a decline in matching ability. The experimental findings demonstrate that a better balance can be achieved with approximately 4 bi-encoder layers when employing a 12-layer pre-trained language model.

5.5 Results on Different Query Sets

LeCaRD divides query cases into two categories: common queries and controversial queries. Generally speaking, controversial queries are more difficult compared to common queries. There are 77 common queries and 30 controversial queries in LeCaRD. We select Elem4LCR-I as the representative of our proposed methods as it shows more obvious advantage in ranking performance. The results, as shown in Figure 6, reveal the following

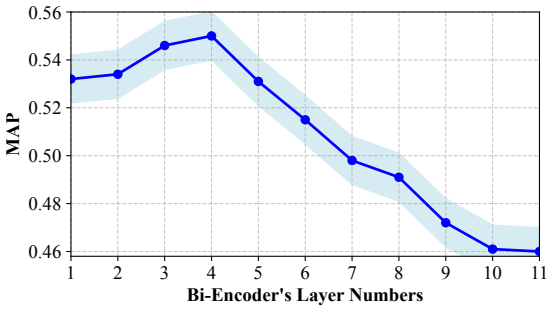

Figure 5: Elem4LCR-I's performance on different bi-encoder layers.

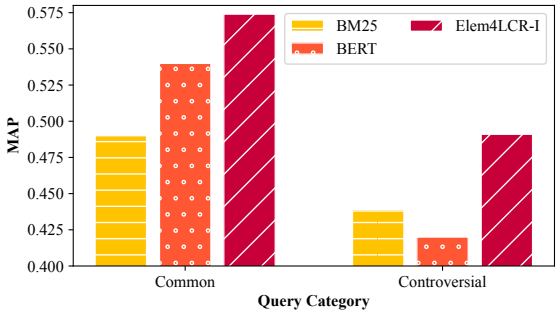

Figure 6: Elem4LCR-I's performance on the common query set and the controversial query set.

observations: (1) Elem4LCR-I outperforms other baselines on both sets, which demonstrates the effectiveness of using legal element information. (2) On the controversial queries, BERT performs even worse than BM25, but Elem4LCR-I still shows an obvious advantage compared with baselines. It shows that expert knowledge of legal elements is effective for solving complicated samples, and this conclusion is consistent with the example we described in Figure 1.

6 Conclusion

In this paper, we contribute a new legal element dataset (i.e., LeCaRD-Elem) by annotating the legal elements of the widely-used LeCaRD dataset using an efficient semi-automatic annotation method. Based on the proposed LeCaRD-Elem dataset, we take the first step to explore the incorporation of legal element knowledge for enhancing legal case retrieval by proposing two legal element-aware ranking models (i.e., Elem4LCR-E and Elem4LCR-I). Experimental results demonstrate superior the ranking performance of our proposed models over existing baselines.

7 Limitations

The legal domain encompasses many specialized subfields (e.g., criminal law and civil law). Similar to most previous research on legal case retrieval, our work focuses on Chinese criminal cases. Besides, the proposed approaches require legal element annotations during training, which restricts their transferability to certain datasets.

8 Ethical Considerations

The legal domain is a sensitive area for the application of NLP technology. Our proposed methods aim to enhance the performance of legal case retrieval systems, yet they can not guarantee uniformly high-quality results for all queries. In real-world scenarios, multiple factors such as out-of-distribution queries and the lack of similar cases in the case corpus can result in poor retrieval performance. Based on the above discussion, we advise expert users to carefully examine the search results and independently determine their suitability for reference.

References

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. 2018. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062*.

Trevor Bench-Capon, Michał Araszkiewicz, Kevin Ashley, Katie Atkinson, Floris Bex, Filipe Borges, Daniele Bourcier, Paul Bourgine, Jack G Conrad, Enrico Francesconi, et al. 2012. A history of ai and law in 50 papers: 25 years of the international conference on ai and law. *Artificial Intelligence and Law*, 20:215–319.

Paheli Bhattacharya, Kripabandhu Ghosh, Saptarshi Ghosh, Arindam Pal, Parth Mehta, Arnab Bhattacharya, and Prasenjit Majumder. 2019. Fire 2019 aila track: Artificial intelligence for legal assistance. In *Proceedings of the 11th Annual Meeting of the Forum for Information Retrieval Evaluation*, pages 4–6.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. Legal-bert: The muppets straight out of law school. *arXiv preprint arXiv:2010.02559*.

NCBI Resource Coordinators. 2016. Database resources of the national center for biotechnology information. *Nucleic acids research*, 44(D1):D7–D19.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

George P Fletcher. 2001. Criminal theory in the twentieth century. *Theoretical inquiries in law*, 2(1).

Hanjo Hamann. 2019. The german federal courts dataset 1950–2019: from paper archives to linked open data. *Journal of empirical legal studies*, 16(3):671–688.

Philip J Hayes and Steven P Weinstein. 1990. Construe/tis: A system for content-based indexing of a database of news stories. In *IAAI*, volume 90, pages 49–64.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2018. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*.

Yoshinobu Kano, Mi-Young Kim, Masaharu Yoshioka, Yao Lu, Juliano Rabelo, Naoki Kiyota, Randy Goebel, and Ken Satoh. 2018. Coliee-2018: Evaluation of the competition on legal information extraction and entailment. In *JSAI International Symposium on Artificial Intelligence*, pages 177–192. Springer.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Daniel Locke and Guido Zuccon. 2018. A test collection for evaluating legal case law search. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1261–1264.

Yougang Lyu, Zihan Wang, Zhaochun Ren, Pengjie Ren, Zhumin Chen, Xiaozhong Liu, Yujun Li, Hongsong Li, and Hongye Song. 2022. Improving legal judgment prediction through reinforced criminal element extraction. *Information Processing & Management*, 59(1):102780.

Yixiao Ma, Yunqiu Shao, Yueyue Wu, Yiqun Liu, Ruizhe Zhang, Min Zhang, and Shaoping Ma. 2021. Lecard: a legal case retrieval dataset for chinese law system. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 2342–2348.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*.

Jay M Ponte and W Bruce Croft. 2017. A language modeling approach to information retrieval. In *ACM SIGIR Forum*, volume 51, pages 202–208. ACM New York, NY, USA.

Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2019. Understanding the behaviors of bert in ranking. *arXiv preprint arXiv:1904.07531*.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at trec-3. *Nist Special Publication Sp*, 109:109.

Guilherme Moraes Rosa, Ruan Chaves Rodrigues, Roberto Lotufo, and Rodrigo Nogueira. 2021. Yes, bm25 is a strong baseline for legal case retrieval. *arXiv preprint arXiv:2105.05686*.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Manavalan Saravanan, Balaraman Ravindran, and Shivani Raman. 2009. Improving legal information retrieval using an ontological framework. *Artificial Intelligence and Law*, 17:101–124.

Yunqiu Shao, Jiaxin Mao, Yiqun Liu, Weizhi Ma, Ken Satoh, Min Zhang, and Shaoping Ma. 2020. Bert-pli: Modeling paragraph-level interactions for legal case retrieval. In *IJCAI*, pages 3501–3507.

Yi Shu, Yao Zhao, Xianghui Zeng, and Qingli Ma. 2019. Cail2019-fe. *Tech. Rep.*

Vu Tran, Minh Le Nguyen, Satoshi Tojo, and Ken Satoh. 2020. Encoded summarization: summarizing documents into continuous vector space for legal case retrieval. *Artificial Intelligence and Law*, 28:441–467.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16(1):1–28.

Joe H Ward Jr. 1963. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244.

Junqiu Wei, Xiaozhe Ren, Xiaoguang Li, Wenyong Huang, Yi Liao, Yasheng Wang, Jiashu Lin, Xin Jiang, Xiao Chen, and Qun Liu. 2019. Nezha: Neural contextualized representation for chinese language understanding. *arXiv preprint arXiv:1909.00204*.

Chaojun Xiao, Xueyu Hu, Zhiyuan Liu, Cunchao Tu, and Maosong Sun. 2021. Lawformer: A pre-trained language model for chinese legal long documents. *AI Open*, 2:79–84.

Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Yansong Feng, Xianpei Han, Zhen Hu, Heng Wang, et al. 2018. Cail2018: A large-scale legal dataset for judgment prediction. *arXiv preprint arXiv:1807.02478*.

Feng Yao, Chaojun Xiao, Xiaozhi Wang, Zhiyuan Liu, Lei Hou, Cunchao Tu, Juanzi Li, Yun Liu, Weixing Shen, and Maosong Sun. 2022. LEVEN: A large-scale chinese legal event detection dataset. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 183–201. Association for Computational Linguistics.

Jie Zhao, Ziyu Guan, Cai Xu, Wei Zhao, and Enze Chen. 2022. Charge prediction by constitutive elements matching of crimes. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI*, volume 22, pages 4517–4523.

Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. How does nlp benefit legal system: A summary of legal artificial intelligence. *arXiv preprint arXiv:2004.12158*.

Haoxi Zhong, Zhengyan Zhang, Zhiyuan Liu, and Maosong Sun. 2019. Open chinese language pre-trained model zoo. Technical report.

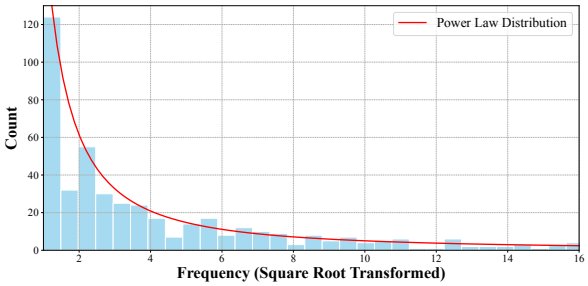

Figure 7: Visualization of the dataset's long-tail distribution.

A LeCaRD-Elem's Long-tail Distribution

As shown in Figure 7, the red curve in the figure represents the power law distribution employed for data fitting. Furthermore, we analyze typical elements sampled from both high-frequency and low-frequency elements. Typical representatives of high-frequency elements are "confessing", "collusion with others" and "voluntary surrender", which may exist in different types of cases due to their general characteristics; Representatives of low-frequency elements are "forging invoices" and "production and sales of inferior pesticides", which are both specific behaviors. This demonstrates that the LeCaRD-Elem dataset exhibits fine-grained characteristics as a result of the two-stage data construction process, providing more exploration space for downstream tasks.

B Training details of Elem4LCR-E

For the ranking task, our goal is to train a model capable of taking the respective legal elements of query and document as input, and then estimating

the relevance between them. The respective legal element set E_q and E_d of the query and the candidate document are formulated as text sequences. The two legal element sequences are concatenated and input into the BERT encoder to learn the relevance feature and score:

$$\begin{aligned} \mathcal{F}_{\text{elem}} &= \text{BERT}_{[\text{CLS}]}(E_q \circ \circ E_d), \\ \mathcal{S}_{\text{elem}} &= \phi(\mathcal{F}_{\text{elem}}), \end{aligned} \tag{8}$$

where $\text{BERT}_{[\text{CLS}]}$ is the embedding output of [CLS] token of BERT encoder, $\circ$ denotes concatenation, $\phi(\cdot)$ is a multi-layer perceptron that transforms relevance feature to a score. Given two documents d_i and d_j, the probability that d_i is more relevant than d_j can be computed as follows:

$$P_{ij} = \frac{1}{1 + \exp(\mathcal{S}_{\text{elem}}^{d_j} - \mathcal{S}_{\text{elem}}^{d_i})}. \tag{9}$$

We denote $\bar{P}_{ij}$ as the real probability. If d_i is more relevant than d_j then $\bar{P}_{ij} = 1$, otherwise $\bar{P}_{ij} = 0$. Finally, we use the cross entropy function to define the pairwise ranking loss:

$$\mathcal{L}_{\text{rank}} = -\bar{P}_{ij} \log P_{ij} - (1 - \bar{P}_{ij}) \log(1 - P_{ij}). \tag{10}$$

C Fundamental Knowledge of Mutual Information

Mutual information is an important tool for quantifying the dependency between two random variables. Mathematically, it is defined as the relative entropy between their joint distribution and marginal distributions:

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X), \end{aligned} \tag{11}$$

where $H(\cdot)$ is the Shannon entropy, X and Y are two random variables. However, in the practice of deep learning, the representation space of random variables is usually very high-dimensional. This brings a great challenge for estimating the mutual information. To address this issue, Belghazi et al. (2018) proposed mutual information neural estimation (MINE), transforming the optimization target to a lower bound based on Donsker-Varadhan representation of KL-divergence:

$$\hat{I}_\theta^{\text{DV}}(X;Y) := \mathbb{E}_{\mathbb{J}}[T_\theta(x,y)] - \log \mathbb{E}_{\mathbb{M}}[e^{T_\theta(x,y)}], \tag{12}$$

where $\mathbb{E}_{\mathbb{J}}$ and $\mathbb{E}_{\mathbb{M}}$ represent joint distribution and marginal distribution respectively, $T_\theta : X \times Y \to \mathbb{R}$ is a neural network approximator. Furthermore, replacing KL-divergence with JS-divergence will lead to a more stable optimization process and better results (Hjelm et al., 2018):

$$\begin{aligned} \hat{I}_\theta^{\text{JSD}}(X;Y) := \mathbb{E}_{\mathbb{J}}[-\text{sp}(-T_\theta(x,y))] \\ - \mathbb{E}_{\mathbb{M}}[\text{sp}(T_\theta(x,y))], \end{aligned} \tag{13}$$

where $\text{sp}(z) = \log(1 + e^z)$ is the softplus function.

D Implementation Details

Elem4LCR-E. In the element prediction task, the learning rate is set to 1e-4. α and γ of focal loss are set to 2 and 0.7, respectively. In the ranking task, we impose a maximum length restriction of 150 tokens for both query and document elements, which can cover all the elements for over 99.8% cases. The learning rate is set to 3e-6.

Elem4LCR-I. The element-based teacher ranker and the description-based student ranker need to be trained. We conducted multiple experiments to select the parameters of these models. For the element-based teacher ranker, the prompt length is 20, the batch size is set to 128, and the learning rate is set to 3e-5; For the description-based student ranker, the lengths of query tokens and case tokens are set to 250 and 259 respectively. The learning rate is set to 1e-5. All experiments are conducted on four Nvidia A100-40g GPUs.

E Discussion of the Two Proposed Approaches

As shown in Table 2, based on the current element prediction accuracy, Elem4LCR-I achieves superior ranking performance. However, it's worth noting that the proposed implicit approach is not consistently more suitable than the explicit one in all scenarios. Specifically, our suggestions are: (1) In scenarios where users directly perform the search (e.g. case retrieval system), the explainability of explicit element labels provided by Elem4LCR-E is a user-friendly advantage. Moreover, users who are expert in law can manually modify the mistaken element labels of the query, which also potentially improves the performance of the explicit approach. (2) When retrieval is only an auxiliary module of the system (e.g. the retrieval module of a large language model), the benefits of better ranking results provided by Elem4LCR-I will be more important.