Learning Interpretable Legal Case Retrieval via Knowledge-Guided Case Reformulation

Anonymous ACL submission

Abstract

Legal case retrieval for sourcing similar cases is critical in upholding judicial fairness. Different from general web search, legal case retrieval involves processing lengthy, complex, and highly specialized legal documents. Existing methods in this domain often overlook the incorporation of legal expert knowledge, which is crucial for accurately understanding and modeling legal cases, leading to unsatisfactory retrieval performance. This paper introduces KELLER, a legal knowledge-guided case reformulation approach based on large language models (LLMs) for effective and interpretable legal case retrieval. By incorporating professional legal knowledge about crimes and law articles, we enable large language models to accurately reformulate the original legal case into concise sub-facts of crimes, which contain the essential information of the case. Extensive experiments on two legal case retrieval benchmarks demonstrate superior retrieval performance and robustness on complex legal case queries of KELLER over existing methods.

1 Introduction

Legal case retrieval is vital for legal experts to make informed decisions by thoroughly analyzing relevant precedents, which upholds justice and fairness (Hamann, 2019). This practice is crucial in both common law and civil law systems globally (Lastres, 2015; Harris, 2002). In civil law, although following past cases (known as "stare decisis") is not mandatory, judges are still highly advised to consider previous cases to improve the accuracy and trustworthiness of their judgments.

In legal case retrieval, both the query and the document are structured legal cases, distinguishing the task from other information retrieval (IR) tasks. Specifically, as shown in Figure 1, a legal case document comprises several sections, such as procedure, facts, and the court's decision, making it much longer than typical queries and passages in

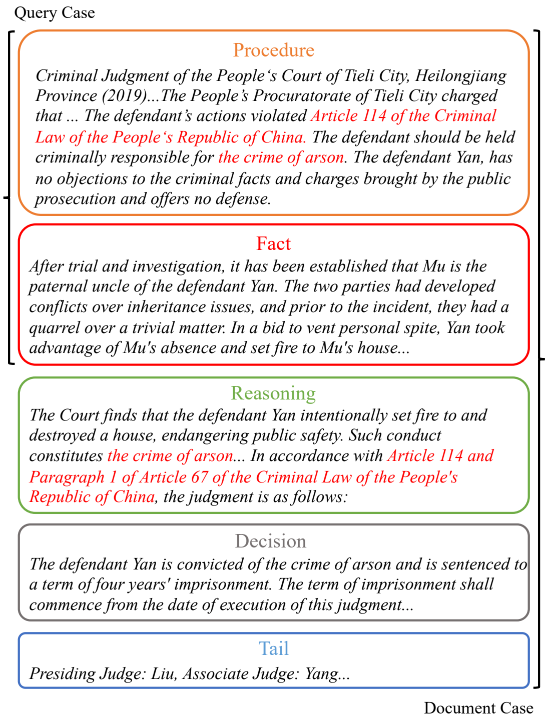

Figure 1: The query case and candidate document case examples. The query case typically contains only partial content since it has not been adjudicated. Extractable crimes and law articles are highlighted in red.

the standard ad-hoc search tasks. Its average text length often exceeds the maximum input limits of popular retrievers, such as 512 tokens (Devlin et al., 2019). Moreover, a legal case may encompass multiple, distinct criminal behaviors. Comprehensively considering all criminal behaviors of a legal case is important in determining its matching relevance with a query case. However, these key criminal descriptions are usually dispersed throughout the lengthy contents, which can significantly affect the effectiveness of traditional long document modeling strategies like FirstP and MaxP (Dai and Callan, 2019) in the legal domain.

To tackle the challenge of comprehending long and complex legal cases, previous works mainly fall into two categories. The first approach focuses on expanding the context window size (Xiao et al., 2021) or splitting legal cases into passages (Shao et al., 2020). However, given the specialized and complex nature of legal texts, merely increasing the context window size still proves insufficient for significantly improving the retrieval performance. In contrast, the second approach performs direct text summarization (Askari and Verberne, 2021; Tang et al., 2023) or embedding-level summarization (Yu et al., 2022) on the legal case, aiming to only keep the most crucial information for assessing the relevance between legal cases. However, they typically only rely on heuristic rules or the models' inherent knowledge for summarization. As the legal domain is highly specialized, existing approaches that overlook professional legal knowledge (e.g., law articles) are likely to perform inaccurate summarization.

In this paper, we present a Knowledge-guidEd case reformuLation approach for LEgal case Retrieval, named KELLER. Our main idea is to leverage professional legal knowledge to guide large language models (LLMs) to summarize the corresponding key sub-facts for the crimes of the legal cases, and then directly learn to model case relevance based on these crucial and concise sub-facts.

Due to the specialization and complexity of the legal case, it is quite challenging to directly summarize the corresponding key sub-facts for all the crimes from the legal case, even using advanced LLMs (Tang et al., 2023). To address this problem, we propose a two-step legal knowledge-guided prompting method, as illustrated in the left side of Figure 2. In the initial step, we prompt LLM to extract all of the crimes and law articles contained in the legal case and then perform post-processing on them to establish correct mappings between the crimes and law articles by referring to the legal expert database. In the next step, we prompt LLM with the extracted "crime-article " pairs to summarize the sub-fact of the crime from the legal case. The intermediate law articles, serving as high-level abstractions of the actual criminal events, can largely reduce the difficulty of identifying the corresponding sub-fact for the crime and improve accuracy. Figure 5 shows an example of three summarized sub-facts from a legal case.

Then, we directly model the case relevance based on these sub-facts because they are not only the most crucial information for relevance judgment in legal case retrieval but are also concise enough to meet the text length limitations of popular pre-trained retrieval models. For the comprehensive consideration of effectiveness, efficiency, and interoperability, we adopt the simple *MaxSim* and *Sum* operators to aggregate the relevance scores between query and document sub-facts to get the final case relevance score. The model is trained with dual-level contrastive learning to comprehensively capture the matching signals at the case level and the sub-fact level. On two widely-used datasets, we show that KELLER achieves new state-of-the-art results in both zero-shot and fine-tuning settings. Remarkably, KELLER demonstrates substantial improvements in handling complex queries.

Our main contributions can be summarized as:

(1) We propose to leverage professional legal knowledge about crimes and law articles to equip LLM with much-improved capabilities for summarizing essential sub-facts from complex cases.

(2) We suggest performing simple *MaxSim* and *Sum* aggregation directly on those refined sub-facts to achieve effective and interpretable legal retrieval.

(3) We introduce dual-level contrastive learning that enables the model to capture multi-granularity matching signals from both case-level and sub-fact-level for enhanced retrieval performance.

2 Related Work

Legal case retrieval. Existing legal case retrieval methods are categorized into statistical and neural models. Statistical models, notably the BM25 algorithm, can be enhanced by incorporating legal expert knowledge such as legal summarization (Tran et al., 2020; Askari and Verberne, 2021), issue elements (Zeng et al., 2005) and ontology (Saravanan et al., 2009). Neural models have been advanced through deep learning and the use of pre-trained language models (Devlin et al., 2019; Zhong et al., 2019; Chalkidis et al., 2020; Zhang et al., 2023). Recent advancements in this domain include the design of specialized pre-training tasks tailored for legal case retrieval, which yields remarkable improvements in retrieval metrics (Li et al., 2023a; Ma et al., 2023b).

Due to the limitations of neural models in handling long texts, researchers mainly focus on processing lengthy legal documents by isolating the "fact description" section and truncating it to fit the model's input constraints (Ma et al., 2021; Yao et al., 2022; Ma et al., 2023b; Li et al., 2023a). To overcome the long-text problem, some other strategies include segmenting texts into paragraphs for interaction modeling (Shao et al., 2020), employing architectures like Longformer for extensive pre-training on legal texts (Xiao et al., 2021), and transforming token-level inputs into sentence-level encoding (Yu et al., 2022).

Query rewriting with LLMs. Recently, researchers naturally employ LLMs to enhance the effectiveness of query rewriting (Zhu et al., 2023; Mao et al., 2023; Ma et al., 2023a; Wang et al., 2023; Jagerman et al., 2023). For instance, HyDE (Gao et al., 2023) creates pseudo passages for better query answers, integrating them into a vector for retrieval, while Query2Doc (Wang et al., 2023) employs few-shot methods to generate precise responses. Furthermore, Jagerman et al. (2023) explores LLMs' reasoning capacities to develop "Chain-of-Thoughts" responses for complex queries. However, the above methods struggle with legal case retrieval, where both queries and documents are lengthy cases. In the legal domain, PromptCase (Tang et al., 2023) attempts to address this by summarizing case facts within 50 words, but this approach often misses important details as many cases feature multiple independent facts.

3 Methodology

In this section, we first introduce some basic concepts in legal case retrieval. Then we delve into the three core parts of our KELLER, including legal knowledge-guided case reformulation, relevance modeling, and dual-level contrastive learning.

3.1 Preliminaries

In legal case retrieval, both queries and candidate documents are real structured legal cases that can extend to thousands of tokens in length. Figure 1 shows an illustration of the typical case structure. Specifically, a case usually contains several sections, including *procedure*, *fact*, *reasoning*, *decision*, and *tail*. Notably, the candidate documents are completed legal cases that have been through the adjudication process and therefore contain all sections. In contrast, the query cases are not yet adjudicated, so they usually only include the *procedure* and *fact* sections.

Formally, given a query case q and a set of document cases D, the objective of legal case retrieval is to calculate a relevance score s between the query case and each document case in D, and then rank the document cases accordingly.

3.2 Knowledge-Guided Case Reformulation

When assessing the relevance between two legal cases, the key facts of their crimes are the most crucial things for consideration. Therefore, given the complexity of the original legal cases which makes direct learning challenging, we try to first refine the legal cases into shorter but more essential "crime-fact" snippets. For example, we can get such a snippet from the case shown in Figure 1, whose crime is *"the crime of arson"* and the fact is *"Yan took advantage of Mu's absence and set fire ..."*.

However, the description of a crime and its corresponding facts are often scattered throughout the lengthy case, and a single case may contain multiple crimes and facts, significantly complicating the extraction process. To tackle this problem, we propose a two-step prompting method leveraging professional legal knowledge to guide LLM to achieve accurate extraction.

Crime and law article extraction. First, we prompt LLM to extract all crimes and all law articles from the case. This step is relatively straightforward for LLM, as each crime and law article is a distinct, identifiable element within the text. For example, the extracted crime and law article for the case shown in Figure 1 are *"the crime of arson"* and *"Article 114 and Paragraph 1 of Article 67 of the Criminal Law of the People's Republic of China"*, respectively. Our extraction prompt is shown in Appendix B.

Post-Processing. The extracted law articles may just be the titles. We then expand these titles into full articles by gathering their detailed provision content from the Web based on the titles. Then, we establish a mapping between each crime and its relevant law articles by referring to a database built by our legal experts. Note that the correlation between specific crimes and their corresponding legal articles is objective, as it is clearly defined by law. After post-processing, we can obtain all the "crime-articles" pairs for a legal case.

Fact summarization. Next, we leverage the extracted crimes and their relevant law articles to guide LLM in summarizing the specific facts of each crime from the original legal case. The law articles, serving as high-level abstractions of the actual criminal events, can considerably simplify

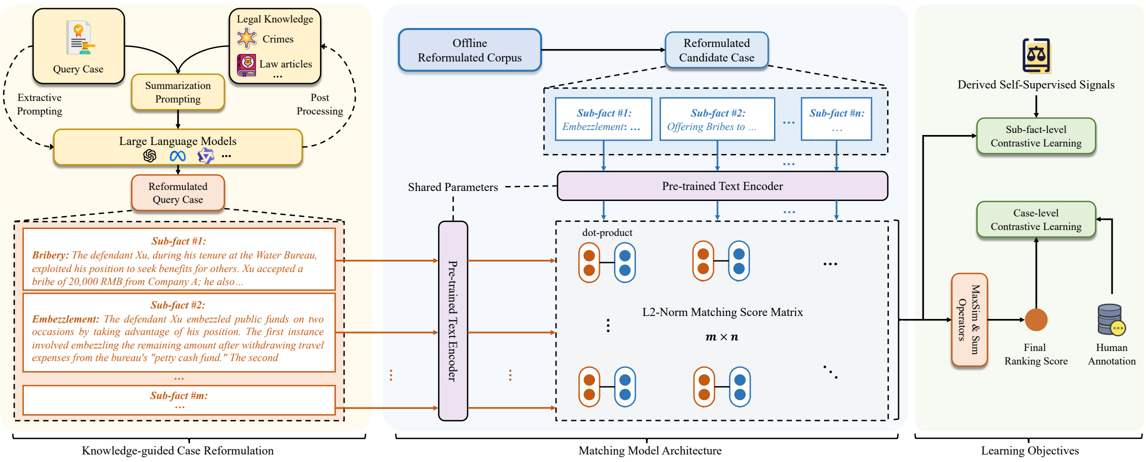

Figure 2: Overview of KELLER. We first perform legal knowledge-guided prompting to reformulate the legal cases into a series of crucial and concise sub-facts. Then, we directly model the case relevance based on the sub-facts. The model is trained at both the coarse-grained case level and the fine-grained sub-fact level via contrastive learning.

the task of identifying the corresponding specific facts. The prompt for fact summarization is shown in Appendix B.2.

Through our legal knowledge-guided reformulation, we can accurately distill a series of crimes and their corresponding specific facts from the originally lengthy legal cases. Finally, we form a *sub-fact* snippet, with the crime as the title and its facts as the main body. These refined sub-facts are not only the most crucial information for relevance judgment in legal case retrieval but are also concise enough to meet the text length limitations of popular pre-trained retrieval models. Please note that, since the required legal knowledge is present in criminal case documents from mainstream countries (e.g., China and the United States), our approach is actually internationally applicable. Our materials in Appendix D further prove this.

3.3 Relevance Modeling

We directly model the relevance of legal cases using the refined sub-facts, rather than relying on the full text of the original legal cases. Specifically, given a query case $q = \{q_1, ..., q_m\}$ and a candidate case $d = \{d_1, ..., d_n\}$, where q_i represents the i-th sub-fact of q and d_j represents the j-th sub-fact of d. We utilize a pre-trained text encoder to encode them:

$$\begin{aligned} E_{q_i} &= \text{Pool}_{\text{[CLS]}}\left(\text{Encoder}(q_i)\right), \\ E_{d_j} &= \text{Pool}_{\text{[CLS]}}\left(\text{Encoder}(d_j)\right), \end{aligned} \tag{1}$$

where $\text{Pool}_{\text{[CLS]}}$ means extracting the embedding output at the [CLS] token position. Then, we compute the similarity matrix $\mathbf{M}_{m \times n}$ using the L2-norm dot product. Each element $M_{i,j}$ of M is the similarity calculated between the normalized embeddings of the i-th sub-fact in the reformulated query case and j-th sub-fact in the reformulated document case:

$$M_{i,j} = \text{Sim}(E_{q_i}, E_{d_j}) = \text{Norm}(E_{q_i}) \cdot \text{Norm}(E_{d_j}^T). \tag{2}$$

Finally, we aggregate this similarity matrix to derive the matching score. There are various sophisticated choices for aggregation, such as using attention or kernel pooling (Xiong et al., 2017). In this paper, we opt to employ the *MaxSim* and *Sum* operators (Khattab and Zaharia, 2020):

$$s_{q,d} = \sum_{i=1}^{m} \text{Max}_{j=1}^{n} M_{i,j}, \tag{3}$$

where $s_{q,d}$ is the final predicted relevance score. We choose these two operators because of their advantages in effectiveness, efficiency, and interpretability over the other aggregation approaches for our scenario:

(1) Effectiveness: Typically, each query's sub-fact q_i matches one document sub-fact d_j at most in practice, which is well-suited for *MaxSim* of applying the Max operation across all document's sub-facts for a given query's sub-fact. For instance, considering a query sub-fact about *"drug trafficking"*, and the document sub-facts about *"drug trafficking"* and *"the discovery of privately stored guns and ammunition"*, only the *"drug trafficking"* sub-fact of the document is relevant for providing matching evidence. In contrast, using soft aggregation methods (e.g., kernel pooling (Xiong et al., 2017)) may introduce additional noise in this scenario.

(2) Efficiency: *Maxsim* and *Sum* operations on tensors are quite efficient for both re-ranking and large-scale top-k retrieval supported by multi-vector-based Approximate Nearest Neighbor algorithms (Khattab and Zaharia, 2020). This high efficiency is important for meeting the low-latency requirements of the practical use.

(3) Interpretability: *MaxSim* provides clear interpretability by revealing the quantitative contribution of each query and document sub-fact towards the final relevance score, which can aid in understanding the ranking strategies and justifying the retrieval results. We further illustrate this advantage by studying a real case in Section 4.6.

3.4 Dual-Level Contrastive Learning

We incorporate matching signals from both the coarse-grained case level and the fine-grained sub-fact level to comprehensively enhance the model performance in legal case matching.

Case-level contrastive learning. At the case level, we consider directly optimizing toward the final matching score between the query case and the document cases. Specifically, we employ the classical ranking loss function to promote the relevance score between the query and the positive document while reducing it for negative documents:

$$\mathcal{L}_{\text{R}} = -\log \frac{\exp(s_{q,d^+}/\tau)}{\exp(s_{q,d^+}/\tau) + \sum_{d^-} \exp(s_{q,d^-}/\tau)}, \tag{4}$$

where d^+ is the positive document of the query q and each d^- is from the in-batch negatives. τ is a temperature parameter.

Sub-fact-level contrastive learning. At the sub-fact level, we incorporate intermediate relevance signals among sub-facts to fine-grainedly enhance the model's effectiveness in understanding sub-facts' content and their matching relationships. However, only the case-level relevance labels are available in the dataset. Naively considering all the sub-fact pairs between the query and the positive documents as positives and all the sub-fact pairs between the query and the negative documents as negatives will introduce substantial false positive and negative noise. To mitigate this issue, we propose a heuristic strategy to obtain high-quality relevance labels for the query's sub-facts $\{q_1, ..., q_m\}$. The core idea of this strategy is to combine the case-level relevance and the charges of each sub-fact to accurately identify true positive and negative samples. We introduce the details of this strategy in Appendix C due to the space limitation.

After getting the sub-fact level relevance labels, we also adopt the ranking loss function for sub-fact level contrastive learning:

$$\mathcal{L}_{\text{S}} = -\log \frac{\exp(s_{M_{i,j^+}}/\tau)}{\exp(s_{M_{i,j^+}}/\tau) + \sum_{J^-} \exp(s_{M_{i,j^-}}/\tau)}, \tag{5}$$

where M_{i,j^+} are the similarity score between q_i and its positive document. M_{i,j^-} are the similarity score between q_i and its negative document sub-fact. J^- is the collection of all negative document sub-facts for q_i. The final learning objective is the combination of $\mathcal{L}_{\text{R}}$ and $\mathcal{L}_{\text{S}}$:

$$\mathcal{L} = \mathcal{L}_{\text{R}} + \alpha\mathcal{L}_{\text{S}}, \tag{6}$$

where α is a hyper-parameter to adjust the weights of two losses.

4 Experiments

4.1 Experimental Setup

Dataset and evaluation metrics. We conduct extensive experiments on two widely-used datasets: LeCaRD (Ma et al., 2021) and LeCaRDv2 (Li et al., 2023b), whose statistics are listed in Appendix A.1. Considering the limited number of queries in LeCaRD, we directly evaluate all the queries of LeCaRD using the best model trained on LeCaRDv2, thereby avoiding the need for dataset split. Following the previous studies (Li et al., 2023a,b), we regard label=3 in LeCaRD and label≥2 in LeCaRDv2 as positive. For the query whose candidate documents are all annotated as positive, we supplement the candidate pool by sampling 10 document cases from the top 100-150 BM25 results. To exclude the effect of unlabeled potential positives in the corpus, we rank the candidate pools and adopt MAP, P@k (k=3), and NDCG@k (k=3, 5, 10) as our evaluation metrics.

Baselines. We compare KELLER against the following baselines across three categories. The first is *traditional probabilistic models*, including TF-IDF and BM25. The second is *ranking methods based on pre-trained language models*, including BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), BGE (Xiao et al., 2023) and SAILER (Li

Table 1: Main results of the fine-tuned setting on LeCaRD and LeCaRDv2. "†" indicates our approach outperforms all baselines significantly with paired t-test at p < 0.05 level. The best results are in bold.

Model	LeCaRD					LeCaRDv2				
	MAP	P@3	NDCG@3	NDCG@5	NDCG@10	MAP	P@3	NDCG@3	NDCG@5	NDCG@10
Traditional ranking baselines										
BM25	47.30	40.00	64.45	65.59	69.15	55.20	48.75	72.11	72.51	79.85
TF-IDF	42.59	36.19	58.14	59.98	63.37	55.19	47.92	71.38	72.70	75.04
PLM-based neural ranking baselines										
BERT	53.83	50.79	73.19	73.43	75.54	60.66	53.12	77.78	78.73	80.85
RoBERTa	55.79	53.33	74.40	74.33	76.70	59.75	53.12	78.15	78.97	80.70
BGE	54.98	53.33	74.29	74.09	75.65	60.64	51.87	76.99	78.43	80.90
SAILER	57.98	56.51	77.55	77.04	79.41	60.62	54.58	78.67	78.99	81.41
Neural ranking baselines designed for long text										
BERT-PLI	48.16	43.80	65.74	68.14	71.32	55.34	46.67	71.62	73.68	76.63
Lawformer	54.58	50.79	73.19	73.43	75.54	60.17	54.17	78.23	78.99	81.40
Case reformulation with LLMs										
PromptCase	59.71	55.92	78.75	78.44	80.71	62.25	54.19	78.51	79.07	81.26
KELLER	**66.84**†	**57.14**	**81.24**†	**82.42**†	**84.67**†	**68.29**†	**63.13**†	**84.97**†	**85.63**†	**87.61**†

et al., 2023a). The third is *ranking methods designed for handling long (legal) text*, including BERT-PLI (Shao et al., 2020), Lawformer (Xiao et al., 2021), and PromptCase (Tang et al., 2023).

Implementations. We introduce the selected language models, hyperparameter settings and other details in Appendix A.2.

4.2 Main Results

The main results are as shown in Table 1 and we have the following observations:

(1) KELLER outperforms all baseline methods across all metrics on both datasets. Compared with previous methods tailored for the long-text problem, KELLER employs knowledge-guided case reformulation to address the challenge of long-text comprehension. This demonstrates the effectiveness of separating comprehension and matching tasks in the domain of legal case retrieval.

(2) After fine-tuning on legal case retrieval datasets, the performance gap between general-purpose and retrieval-oriented PLMs becomes less distinct. This observation may stem from two reasons. First, the scarcity of training data in the legal case retrieval task can induce overfitting to annotation signals, which hampers the model's generalization capabilities. Second, Naive truncation of lengthy texts can make the model's inputs lose sufficient matching signals, leading to inconsistencies between relevance annotations and matching evidence.

(3) We observe that these long-text-oriented baseline methods do not show significant advantages. Despite BERT-PLI and Lawformer processing more text than other methods, their input capacity was still insufficient for the average length of legal cases. Handling both long-text processing and complex semantic understanding within one retriever presents a significant challenge. To address this issue, our approach offloads a portion of the long-text comprehension task via knowledge-guided case reformulation and improves the ranking performance.

4.3 Zero-shot Evaluation

Considering the inherent data scarcity problem in legal case retrieval, we evaluate the zero-shot performance (i.e., without fine-tuning on the training set of LeCaRDv2) of models on LeCaRDv2.

Results are shown in Table 2 and we find that KELLER consistently outperforms baselines in both zero-shot and fine-tuning settings. Upon comparing the performance of each method under zero-shot and fine-tuned settings, we observe that most methods benefit from fine-tuning except SAILER. Intuitively, models trained in a general domain or task could be enhanced through fine-tuning. In specific domains, continued fine-tuning of models generally does not lead to a significant decrease in performance. We posit that the unexpected outcomes in the SAILER model primarily arise from overfitting the limited data used for fine-tuning, which impairs the generalization capabilities established in the pre-training phase.

4.4 Ablation Study

We design the following six ablations: (1) *KGCR→NS*: We replace our Knowledge-Guided Case Reformulation (KGCR) with a Naive Sum-

Table 2: Zero-shot performance on LeCaRD and LeCaRDv2. "†" indicates our approach outperforms all baselines significantly with paired t-test at $p < 0.05$ level. The best results are in bold.

Model	LeCaRD					LeCaRDv2				
	MAP	P@3	NDCG@3	NDCG@5	NDCG@10	MAP	P@3	NDCG@3	NDCG@5	NDCG@10
General PLM-based baselines										
BERT	42.92	37.78	60.11	61.37	64.10	56.46	52.08	75.82	77.05	79.39
RoBERTa	51.50	47.62	69.21	71.07	73.60	57.89	52.08	75.48	76.33	78.38
Lawformer	42.80	38.41	59.46	61.61	64.13	55.05	49.58	74.42	74.31	76.96
Retrieval-oriented pre-training baselines										
BGE	51.81	47.62	68.57	69.91	72.61	57.21	50.42	73.59	75.36	77.80
SAILER	60.62	56.19	79.93	78.99	81.41	62.80	55.00	79.38	81.17	83.83
KELLER	**64.17**†	**57.78**	**80.47**	**81.43**†	**84.36**†	**65.87**†	**61.67**†	**83.33**†	**83.75**†	**86.06**†

marization (NS), which produces case summaries without hierarchical structure. We subsequently optimize the dual encoders with this text as the input. (2)*MS* $\rightarrow$ *Mean*: We replace *MaxSim* and *Sum* (MS) with *Mean* to capture the average relevance of each sub-fact in the candidate cases to the query. (3) *MS* $\rightarrow$ *NC*: We Naively Concatenate (NC) all the reformulated sub-facts into a text sequence and subsequently optimize the dual-encoders. (4) *MS* $\rightarrow$ *KP*: We employ kernel pooling (Xiong et al., 2017) on the score matrix to capture relevance signals. (5) *w/o sfCL*: Training without the sub-fact-level contrastive learning. (6) *w/o SfCL*: Training without the case-level contrastive learning.

Results are shown in Table 3 and we can observe:

(1) Every ablation strategy results in a decline in the model's performance, demonstrating the effectiveness of each module within KELLER. This outcome indicates that KELLER's architecture is both comprehensive and synergistic, with each module contributing to the model's overall performance.

(2) The replacement of the KGCR module exhibits the most significant impact on performance. This highlights the pivotal role of the KGCR module in KELLER. The KGCR module decomposes cases into structured sub-facts, which are crucial for the model's learning process.

(3) Among different aggregation strategies, *MS* $\rightarrow$ *Mean* demonstrates the least performance degradation. This is primarily because the dataset mainly consists of simple cases with single charges, where *Mean* and *MS* become essentially equivalent. Conversely, *MS* $\rightarrow$ *NC* exhibits the most notable performance decline. This is mainly because the model no longer maintains a cross-matching architecture after the concatenation operation. Merging multiple facts into a single representation negatively impacts representation learning.

Table 3: Results of ablation study on LeCaRDv2.

Strategy	MAP	P@3	NDCG@3	NDCG@5	NDCG@10
Effect of knowledge-guided case reformulation					
KGCR$\rightarrow$*NS*	61.91	55.13	79.50	79.11	81.47
Effect of different aggregation strategy					
MS$\rightarrow$*Mean*	67.15	61.81	81.58	84.42	86.74
MS$\rightarrow$*NC*	63.35	57.92	80.37	81.99	84.04
MS$\rightarrow$*KP*	65.47	60.06	79.87	83.61	85.39
Effect of contrastive learning					
w/o SfCL	67.39	61.93	81.24	84.73	86.91
w/o CaCL	67.18	61.67	82.76	84.45	86.51
KELLER	**68.29**	**63.13**	**84.97**	**85.63**	**87.61**

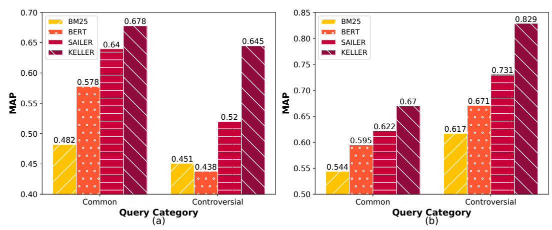

Figure 3: Evaluation on different query types. We evaluate four models on (a) LeCaRD and (b) LeCaRDv2.

4.5 Evaluations on Different Query Types

We investigate the two query types presented in both LeCaRD and LeCaRDv2: *common* and *controversial*. Common queries are similar to initial trials, and controversial queries to retrials, which are typically more complex and require additional expert review. We evaluated multiple models on these query types. Notably, SAILER's performance declined after fine-tuning, so we included its zero-shot results for comparison, alongside the fine-tuned outcomes of other models. Results as shown in Figure 3 and we find:

(1) KELLER outperformed other models on both query types, showing more substantial gains in controversial queries with improvements of 24.04% and 13.41% in the LeCaRD and LeCaRDv2

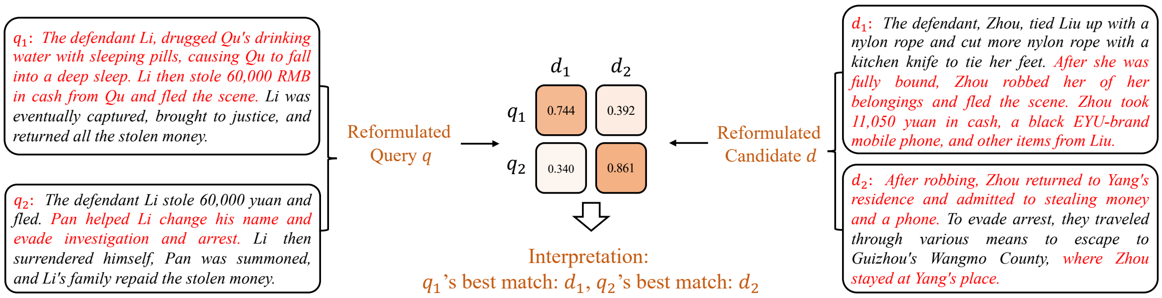

Figure 4: An example of the interpretability of KELLER. We can observe that each sub-fact of the query finds a correct match in the candidate document (in red).

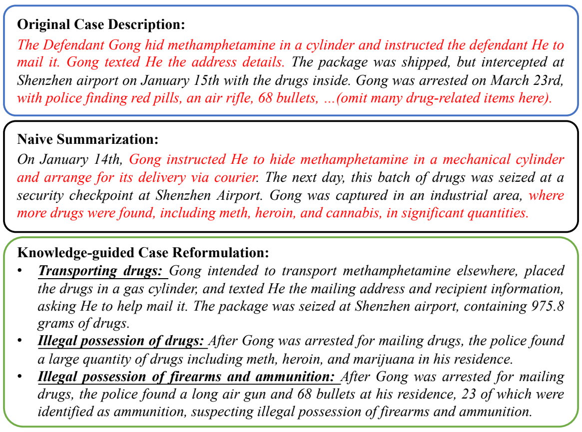

Figure 5: Comparison of the original text, naive summarization, and our proposed knowledge-guided case reformulation. The original text is manually abbreviated due to its length. Important sentences are marked in red.

datasets, respectively. This enhanced performance is credited to KELLER's novel case reformulation, which simplifies complex scenarios into sub-facts, aiding in better comprehension and matching.

(2) In the LeCaRD dataset, lexical-based models showed consistent performance across different queries, unlike representation-based models which varied significantly. For example, BERT outperformed BM25 on common queries but was less effective on controversial ones, a difference attributed to the models' limited ability to handle multifaceted cases. KELLER's cross-matching architecture successfully addresses this limitation.

4.6 Case Studies

Case reformulation. We provide an illustrative comparison between the original case description, naive summarization, and our knowledge-guided case reformulation in Figure 5. The case centers on complex issues of drug transport and firearm possession. Most details focus on drug transportation, with brief mentions of firearms found at the defendant's residence towards the end. Given the 512-token limit of most retrievers, crucial information about the firearms is often inaccessible. While naive summarization captures the main points, it overlooks specifics about the firearms in the context of drug offenses. In contrast, our KGCR method segments the case into three topics—drug transportation, illegal drug possession, and illegal firearms possession—thus detailing each criminal aspect comprehensively.

Interpretability. In KELLER, each sub-fact in a query represents a specific intent of the query, with the highest match score from a candidate case indicating how well this intent is met. KELLER allows users to see which sub-fact in a candidate case matches their intent. For example, in a case involving robbery and harboring crimes shown in Figure 4, KELLER accurately matches sub-facts in the query to those in the candidate case, demonstrating the alignment of KELLER's scoring with the underlying legal facts of the case. The matching is shown in a matrix, where the positions (q_1, d_1) and (q_2, d_2) highlight the defendant's actions in the query and the candidate case, respectively, establishing a direct correlation between the computed scores and the case ranking.

5 Conclusion

In this paper, we introduce KELLER, a ranking model that effectively retrieves legal cases with high interpretability. KELLER structures legal documents into hierarchical texts using LLMs and determines relevance through a cross-matching module. Our tests on two expert-annotated datasets validate its effectiveness. In the future, we will enhance KELLER by incorporating additional specialized knowledge and generative models to refine performance and produce language explanations.

6 Limitations

External Knowledge base Construction. Our method requires constructing a legal knowledge base to assist in case reformulation, which introduces an extra step compared to the out-of-the-box dense retrievers. This issue is common in most domain-specific knowledge-enhanced methods.

Computing Efficiency. Our approach needs to call large language models when processing the query case, which may bring additional computational costs. In our experiments, we have employed techniques such as vLLM to achieve high-speed inference. Furthermore, we believe that with ongoing advancements in techniques in both hardware and algorithms, the computational of utilizing LLMs for processing individual query cases online will be acceptable. For example, Llama3-8B can achieve a speed exceeding 800 tokens per second on the Groq platform, while recent inference services provided by Qwen and DeepSeek require less than $0.0001 per 1,000 tokens.

7 Ethical Discussion

The application of artificial intelligence in the legal domain is sensitive, requiring careful examination and clarification of the associated ethical implications. The two datasets utilized in our experimental analysis have undergone anonymization processes, particularly with regard to personally identifiable information such as names.

Although KELLER demonstrates superior performance on two human-annotated datasets, its recommendations for similar cases may sometimes be imprecise when dealing with intricate real-world queries. Additionally, the case databases in existing systems may not consistently include cases that fully satisfy user requirements. The choice to reference the retrieved cases should remain at the discretion of the experts.

References

Arian Askari and Suzan Verberne. 2021. Combining lexical and neural retrieval with longformer-based summarization for effective case law retrieval. In *Proceedings of the Second International Conference on Design of Experimental Search & Information REtrieval Systems, Padova, Italy, September 15-18, 2021*, volume 2950 of *CEUR Workshop Proceedings*, pages 162–170. CEUR-WS.org.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: the muppets straight out of law school. *CoRR*, abs/2010.02559.

Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for ir with contextual neural language modeling. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 985–988.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1762–1777. Association for Computational Linguistics.

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.

Bruce V Harris. 2002. Final appellate courts overruling their own" wrong" precedents: the ongoing search for principle. *Law Quarterly Review*, 118(July 2002):408–427.

Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. 2023. Query expansion by prompting large language models. *CoRR*, abs/2305.03653.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Steven A Lastres. 2015. Rebooting legal research in a digital age.

Haitao Li, Qingyao Ai, Jia Chen, Qian Dong, Yueyue Wu, Yiqun Liu, Chong Chen, and Qi Tian. 2023a. SAILER: structure-aware pre-trained language model for legal case retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 1035–1044. ACM.

Haitao Li, Yunqiu Shao, Yueyue Wu, Qingyao Ai, Yixiao Ma, and Yiqun Liu. 2023b. Lecardv2: A large-scale chinese legal case retrieval dataset. *arXiv preprint arXiv:2310.17609*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023a. Query rewriting for retrieval-augmented large language models. *CoRR*, abs/2305.14283.

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.

Yixiao Ma, Yueyue Wu, Weihang Su, Qingyao Ai, and Yiqun Liu. 2023b. Caseencoder: A knowledge-enhanced pre-trained model for legal case encoding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 7134–7143. Association for Computational Linguistics.

Kelong Mao, Zhicheng Dou, Fengran Mo, Jiewen Hou, Haonan Chen, and Hongjin Qian. 2023. Large language models know your contextual search intent: A prompting framework for conversational search. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 1211–1225. Association for Computational Linguistics.

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.

Yanran Tang, Ruihong Qiu, and Xue Li. 2023. Prompt-based effective input reformulation for legal case retrieval. In *Databases Theory and Applications - 34th Australasian Database Conference, ADC 2023, Melbourne, VIC, Australia, November 1-3, 2023, Proceedings*, volume 14386 of *Lecture Notes in Computer Science*, pages 87–100. Springer.

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.

Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 9414–9423. Association for Computational Linguistics.

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.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighof. 2023. C-pack: Packaged resources to advance general chinese embedding. *CoRR*, abs/2309.07597.

Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 55–64. ACM.

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.

Weijie Yu, Zhongxiang Sun, Jun Xu, Zhenhua Dong, Xu Chen, Hongteng Xu, and Ji-Rong Wen. 2022. Explainable legal case matching via inverse optimal transport-based rationale extraction. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 657–668. ACM.

Yiming Zeng, Ruili Wang, John Zeleznikow, and Elizabeth A. Kemp. 2005. Knowledge representation for the intelligent legal case retrieval. In *Knowledge-Based Intelligent Information and Engineering Systems, 9th International Conference, KES 2005, Melbourne, Australia, September 14-16, 2005, Proceedings, Part I*, volume 3681 of *Lecture Notes in Computer Science*, pages 339–345. Springer.

Kun Zhang, Chong Chen, Yuanzhuo Wang, Qi Tian, and Long Bai. 2023. Cfgl-lcr: A counterfactual graph learning framework for legal case retrieval. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3332–3341.

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

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Zhicheng Dou, and Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. *CoRR*, abs/2308.07107.

Table 4: Basic statistics of the datasets.

Dataset	LeCaRD	LeCaRDv2
# Train queries	-	640
# Test queries	107	160
# Documents	9,195	55,192
Average query length	445	4,499
Average doc length	7,446	4,768
Average golden docs / query	10.39	13.65

A More Details for Experimental Setup

A.1 Datasets

The statistics of both datasets are listed in Table 4. LeCaRD comprises 107 queries and 10,700 candidate cases. LeCaRDv2, a more extensive collection, includes 800 queries and 55,192 candidate cases.

A.2 Implementation Details

For baseline models, we employ the default parameter settings of Okapi-BM25 in the implementation of BM25. For ranking methods based on PLMs, a uniform learning rate of 1e-5 and a batch size of 128 are consistently applied. In BERT-PLI, the numbers of queries and candidate case segments are set to 3 and 4, respectively, with a maximum segment length of 256. For Lawformer, the maximum text input length is set to 3,072, optimized using a learning rate of 1e-5 and a batch size of 64.

In KELLER, we employ the Qwen-72B-Chat (Bai et al., 2023), which is currently one of the best open-source Chinese LLMs, to perform case reformulation. We do not choose OpenAI API due to concerns about reproducibility and high cost. All prompts, except for the case description, are input as system prompts. In the ranking model, the maximum number of crimes per case is capped at 4, which meets the needs of most cases. We adopt the pre-trained retriever SAILER as the text encoder. The τ in the contrastive learning is 0.01, and the α in the final loss function is 0.9. We conduct model training with a learning rate of 1e-5 and a batch size of 128. All experiments are conducted on four Nvidia Tesla A100-40G GPUs. All the source code and data will be shared at `https://github.com/hide-for-blind-review` if the paper gets accepted.

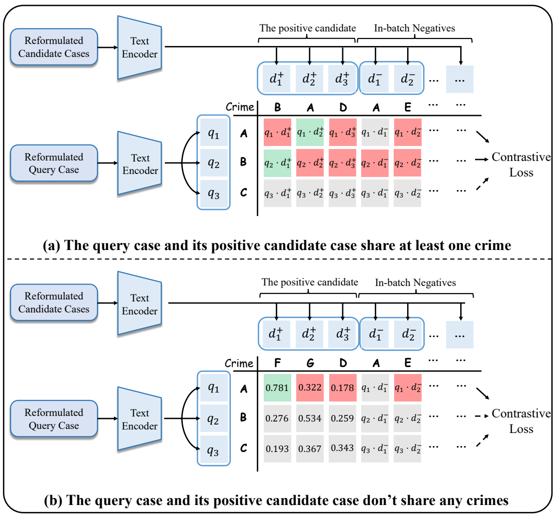

Figure 6: Illustration of our proposed sub-fact-level contrastive learning. The green and red squares represent the positive pairs and negative pairs, respectively. The gray squares are the discarded pairs that are not used for training. The blue rounded rectangles encompass blue squares belonging to the same query/document case. {A, ..., G} are crimes.

B Prompts

B.1 Extraction Prompt

Extraction Prompt: *You are now a legal expert, and your task is to find all the crimes and law articles in the procuratorate's charges (or court judgments) from the provided case. The output format is one line each for crimes and law articles, two lines in total. Multiple crimes (law articles) are separated by semicolons.*

B.2 Summarization Prompt

Summarization Prompt: *You are now a legal expert, and you are good at analyzing lengthy legal case texts containing multiple circumstances of crime. Your task is to concisely summarize the causes, procedures, and outcomes associated with a specified crime, ensuring each part does not exceed 100 words.*
[*Crime*]: *the specific crime name*
[*Law Articles*]: *the specific provisions of law articles*

C Strategy to Obtain Sub-Fact-Level Relevance Labels

Specifically, **for a positive document** d^+ of query q, we first check whether any of the document sub-facts share the same crimes as any of the query sub-facts:

- If it exists, as shown in Figure 6(a), for a query sub-fact q_i, we treat the document sub-facts that share the same crime as the positives (e.g., the green rectangles in columns d_1^+, d_2^+, and d_3^+), and all the other document sub-facts as negatives (e.g., the red rectangles in columns d_1^+, d_2^+, and d_3^+). If the crime of q_i is different from any of the document sub-facts, we will not include q_i for training (e.g., the gray rectangles in row q_3).

- If not, as shown in Figure 6 (b), we select the (q_i, d_j^+) which has the highest similarity score as a positive training pair (e.g., the green rectangle), and retain any $(q_i, d_k^+(k \neq j))$ as negatives (e.g., the red rectangles in columns d_2^+ and d_3^+). All the other query and document sub-fact pairs are discarded (e.g., the gray rectangles in columns d_1^+, d_2^+, and d_3^+).

Then, **for a negative document** d^- of one query sub-fact q_i, we first check whether q_i has one positive sample.

- If not, we discard all the document sub-facts because there doesn't exist a positive sample for contrastive learning (e.g., the gray rectangles of row q_3 in Figure 6 (a) and (b)).

- If it exists, we further check whether one of its document sub-facts d_j^- shares the same crime as a q_i.

  1. Both d_j^- and q_i are implicated to the same crime. we will include all $(q_i, d_k^-(k \neq j))$ as negatives (e.g., the red rectangles of column d_1^- and d_2^- in Figure 6 (a) and (b)). All the other sub-facts are discarded to avoid introducing false negatives (e.g., the gray rectangles of (q_1, d_1^-) in Figure 6 (a) and (b)).
  2. None of d_j^- and q_i pertain to the same crime. We will include all (q_i, d_j^-) as negatives (e.g., the red rectangles of (q_2, d_1^-) and (q_2, d_2^-) in Figure 6 (a)).

D Case Format of Other Regions

To demonstrate the international applicability of our method, we use U.S. legal documents as examples. Figure 7 and Figure 8 depict the formats of a U.S. indictment and a judgment document, respectively. It is evident that the legal knowledge required by our method (a combination of charges and law articles in this paper) is commonly present in the body sections of these documents. our method can be applied to reformulate legal texts in documents from other jurisdictions similarly, thereby enhancing their performance of legal case retrieval.

Indictment Document

Caption

The caption of the case, including the name of the court, the jurisdiction, the title of the case (e.g., "United States v. John Doe"), and the case number.

Introduction

A statement indicating that the grand jury charges the defendant with specific offenses.

Body

Counts:
- Each count of the indictment, specifying the statute the defendant is alleged to have violated.
- A clear and concise statement of the essential facts constituting the offense charged.
- Specific dates, locations, and nature of the criminal acts.

Penalties:
- A section outlining the possible penalties for each count, including fines, imprisonment, and other consequences.

Signatures:
- The signature of the grand jury foreperson.
- The signature of the prosecuting attorney.

Figure 7: Illustration of the indictment document of US.

Judgment Document

Caption

The caption of the case, including the name of the court, the jurisdiction, the title of the case (e.g., "United States v. John Doe"), and the case number.

Introduction

A statement summarizing the trial or plea, the defendant's plea, and the verdict or finding.

Body

Charges and Convictions:
- Listing of each count the defendant was convicted of, with corresponding statute references.

Sentencing:
- Detailed information on the sentence for each count, including imprisonment, supervised release, probation, fines, restitution, and special assessments.
- Conditions of supervised release or probation, if applicable.

Additional Orders:
- Any additional orders, such as forfeiture, asset seizure, or specific directives from the court.

Signatures:
- The signature of the presiding judge.
- The date of the judgment.

Figure 8: Illustration of the judgment document of US.