UniLR: Unleashing the Power of LLMs on Multiple Legal Tasks with a Unified Legal Retriever

Ang Li[1], **Yiquan Wu**[2†], **Yifei Liu**[3], **Ming Cai**[1†], **Lizhi Qing**[5], **Shihang Wang**[5]
Yangyang Kang[1,4], **Chengyuan Liu**[1], **Fei Wu**[1], **Kun Kuang**[1†]

[1]College of Computer Science and Technology, Zhejiang Unirersity, [2]Guanghua Law School, Zhejiang University,
[3] College of Software Technology, Zhejiang University, [4]Polytechnic Institute, Zhejiang University,
[5]Alibaba Group, Hangzhou, China
{leeyon, wuyiquan, liuyifei, cm, yangyangkang, liucy1, wufei, kunkuang}@zju.edu.cn
{yekai.qlz, wangshihang.wsh}@alibaba-inc.com

Abstract

Despite the impressive capabilities of LLMs, they often generate content with factual inaccuracies in LegalAI, which may lead to serious legal consequences. Retrieval-Augmented Generation (RAG), a promising approach, can conveniently integrate specialized knowledge into LLMs. In practice, there are diverse legal knowledge retrieval demands (e.g. law articles and similar cases). However, existing retrieval methods are either designed for general domains, struggling with legal knowledge, or tailored for specific legal tasks, unable to handle diverse legal knowledge types. Therefore, we propose a novel **Uni**fied **L**egal **R**etriever (UniLR) capable of performing multiple legal retrieval tasks for LLMs. Specifically, we introduce attention supervision to guide the retriever in focusing on key elements during knowledge encoding. Next, we design a graph-based method to integrate meta information through a heterogeneous graph, further enriching the knowledge representation. These two components work together to enable UniLR to capture the essence of knowledge hidden beneath formats. Extensive experiments on multiple datasets of common legal tasks demonstrate that UniLR achieves the best retrieval performance and can significantly enhance the performance of LLM.

1 Introduction

Legal artificial intelligence (LegalAI) (Zhong et al., 2020a) focuses on applying artificial intelligence to benefit legal tasks (Zhong et al., 2018, 2020b; Wu et al., 2020, 2022). Recently, the focus has shifted towards leveraging large language models (LLMs) to enhance legal task performance (Fei et al., 2024). However, LLMs still struggle with factual inaccuracies (Mallen et al., 2023; Min et al., 2023), which can lead to serious legal consequences. A promising solution is Retrieval-Augmented Generation (RAG) (Ram et al., 2023), where a retriever fetches relevant knowledge from an external corpus and combines it with the query to generate a more accurate response.

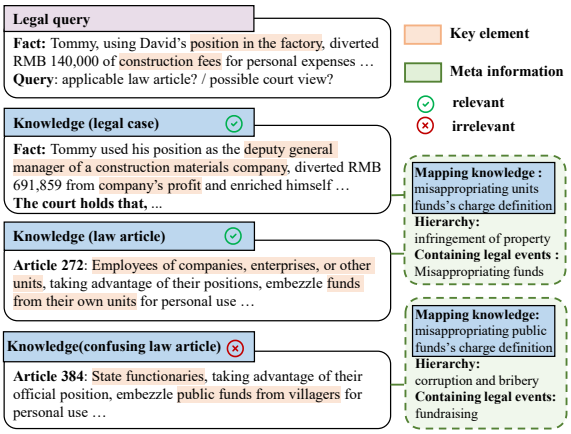

Figure 1: A real-world legal example. For a fact description, if the query asks for applicable articles, relevant articles should be retrieved; if it asks for a possible court view, similar cases should be retrieved for reference.

Legal knowledge has various types such as charge definition, law articles, similar cases, and so on (Burton, 2007). As shown in Fig. 1, in LegalAI practice, addressing various legal tasks requires retrieving different types of legal knowledge. Generally, existing retrieval methods can be divided into two types: Sparse retrieval, like BM25 (Robertson and Zaragoza, 2009) and TF-IDF (Sparck Jones, 1972), can be applied to multiple tasks but struggle with processing complex legal knowledge. Dense retrieval, while effective for specific legal tasks through fine-tuning, cannot handle diverse retrieval needs (Wang et al., 2018, 2019a; Li et al., 2023a). Therefore, it is meaningful to develop a unified retriever that can address multiple legal retrieval tasks.

However, the main challenge in implementing such a retriever lies in the legal knowledge format

† Corresponding Author.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 11953–11967

problem. As shown in Fig. 1, the challenge manifests in two aspects: (1) **Diverse formats across different knowledge types.** For example, law articles are filled with concise legal terminology, while relevant cases are lengthy fact descriptions, which hinders the retriever's understanding. (2) **Subtle differences within the same knowledge type.** In the legal domain, differing only in a few words can yield markedly different legal implications, which may trigger the retriever's confusion. Notably, our findings show that, for the query-relevant knowledge, despite format varying, **key elements**[1] (e.g., defendant's identity, behavior's target) are similar, and **meta information**[1] (e.g., mapping knowledge, hierarchy, legal event schema) remains consistent. Irrelevant but confusing knowledge can also be distinguished using these concepts.

Based on this insight, we propose UniLR, a novel **Uni**fied **L**egal **R**etriever for handling multiple legal retrieval tasks by leveraging key elements and meta information. First, we develop a knowledge mining process. For key elements, we distill expertise from a legal model (Yao et al., 2022), using its attention distribution to identify key elements. For meta information, we construct a heterogeneous knowledge graph where legal knowledge, hierarchical structures, and legal event schemas are represented as different node types, with inter-type and cross-type relationships established based on carefully crafted edge construction rules.

To leverage the mined key elements and meta information, based on a dense embedding model, we design two innovative components: Key Element Supervisor (KES) and Graph-based Knowledge Augmenter (GKA). KES leverages recorded attention and a supervision loss to align the model's focus on key elements. GKA combines graph attention and convolution to aggregate meta information from the graph, enhancing knowledge representation. Finally, we train UniLR using contrastive learning.

Multiple datasets for common legal tasks are experimented in this paper, including charge prediction, law article prediction, court view generation, and legal question answering. Empirical results demonstrate that UniLR achieves the best retrieval performance and significantly enhances LLM's performance. To summarize, our contributions are:

- We investigate multiple legal retrieval tasks for RAG in the LLM era, considering the diverse formats and knowledge confusion.
- We define key elements and meta information of legal knowledge and mine them through expertise distillation and heterogeneous knowledge graph construction, respectively.
- We propose UniLR, a Unified Legal Retriever with two innovative components: KES introduces attention supervision to guide the retriever in focusing on key elements and GKA combines graph attention and convolution to aggregate meta information.
- Extensive experiments on multiple datasets of common legal tasks demonstrate that UniLR achieves the best retrieval performance and significantly enhances LLM capabilities. All data and code are publicly available [2].

2 Related Work

2.1 Legal Artificial Intelligence

In recent years, researchers have focused on using NLP technology to solve specific tasks in the legal field, such as charge prediction (Zhong et al., 2018; Yang et al., 2019; Xu et al., 2020; Yue et al., 2021a; Wu et al., 2022; Chalkidis et al., 2020), article recommendation (Chen et al., 2013; Raghav et al., 2016; Louis and Spanakis, 2022), case retrieval (Raghav et al., 2016; Shao et al., 2020; Li et al., 2023a), court view generation (Wu et al., 2020; Yue et al., 2021b; Li et al., 2024b; Liu et al., 2024), and legal question answering (Zhong et al., 2020b; Kien et al., 2020; Louis et al., 2024). Recently, with the development of LLMs, researchers have transformed various legal tasks into question-and-answer pairs to fine-tune the LLMs, hoping to build unified legal LLMs to solve problems (Cui et al., 2023a; Liu et al., 2023; Huang et al., 2023). However, due to the lack of domain knowledge, both universal LLMs and legal LLMs perform poorly on some legal tasks. Previous work enhances legal judgment prediction and court view generation through various forms of knowledge injection, such as charge definitions, document templates, and reasoning rules (Li et al., 2024a; Zhou et al., 2024a; Li et al., 2025). Because LLM can read text form knowledge directly, a promising approach is enhancing LLMs through retrieval. In the legal

[1]The detailed definitions and acquisition of "key elements" and "meta information" are provided in Sec. 3.2

[2]https://github.com/LIANG-star177/ULKR

domain, researchers have explored retrieval for legal cases and articles(Li et al., 2023a; Wang et al., 2019a), yet there lacks a unified retriever for legal knowledge.

2.2 LLMs and Retrieval

LLMs can learn the knowledge in the retrieved information, which has been validated on many LLMs, such as GPT-3 (Brown et al., 2020), GPT-Neo (Black et al., 2021), and LLaMA (Touvron et al., 2023). In traditional retrieval methods, researchers generally use the BM25 algorithm or dense retrievers based on pre-trained models to retrieve from the training set (Liu et al., 2022; Rubin et al., 2022; Izacard et al., 2022). Recently, researchers have studied the harmonious integration of large models and retrieval (Li et al., 2023b; Luo et al., 2023; Lv et al., 2025). Ge et al. (2023) retrieves knowledge from diverse sources to improve query representation. Ma et al. (2023) uses reinforcement learning to train a query optimizer for semantic alignment. Wang et al. (2023) incorporates contextual recall in pre-training to familiarize models with RAG patterns. However, compared to general retrieval, legal knowledge presents diverse formats and potential confusion, requiring domain-specific solutions. This paper proposes a unified legal knowledge retriever to enhance LLMs in legal tasks.

3 Methodology

In this section, we first formalize the RAG problem for utilizing LLMs in legal tasks and then introduce our UniLR by detailing two parts: Knowledge Mining and Model Architecture. The overall approach of UniLR is illustrated in Fig. 2.

3.1 Problem Formulation

In this paper, we aim to develop a unified retriever that can perform different legal retrieval tasks. We first formulate the problem as follows:

Given a test example (q, y) in task t, where q is a legal query, y is the true label, and the specific knowledge corpus $\mathcal{C}_t = \{c_1, ..., c_{n_c}\}$ consisting of n_c knowledge entries, the probability of the LLM generating the target y based on the query q is defined as follows:

$$p(y|q) = LLM(y|T(c_1, ..., c_k; q)) \quad (1)$$

Here, k is the number of retrieved knowledge entries, T is the template for packaging the retrieved knowledge and the query. We expect the retriever to seamlessly adapt to task transitions.

3.2 Knowledge Mining

3.2.1 Key elements extraction

According to previous work, legal knowledge often describes one or more events (Shen et al., 2020; Li et al., 2020). This paper's key elements correspond to the subjects, objects, and triggers that constitute a legal event. LEVEN (Yao et al., 2022) is a specialized legal event dataset, which groups legal facts into 108 event types, annotating them based on subjects, objects, and trigger words. We observe that while the event prediction model trained on LEVEN places high attention on key elements, it is not well-suited for retrieval tasks. Therefore, we distill the expertise of DMBERT (Wang et al., 2019b), which has the best event prediction performance in the LEVEN.

Specially, given a piece of legal knowledge entry c, we input it into the DMBERT to obtain the attention matrix A of the final layer. Then the attention distribution for the i-th token $\hat{p}_i$ is calculated:

$$\hat{p}_i = \text{softmax}(A_i) = \frac{\sum_{j=1}^{l_c} \exp(A_{ij})}{\sum_{i=1}^{l_c}\sum_{j=1}^{l_c} \exp(A_{ij})} \quad (2)$$

Here, l_c is the knowledge length. Finally, we obtain the attention distribution $\hat{P} = \{\hat{p}_1, ..., \hat{p}_{l_c}\}$ and construct a key element dictionary $\mathcal{D} = \{(c_1 : \hat{P}_1), ..., (c_{n_c} : \hat{P}_{n_c})\}$, where knowledge entries serve as keys and attention distributions as values. In this way, key elements from all corpus are extracted in the form of attention.

3.2.2 Knowledge graph construction

A piece of legal knowledge is intricately linked to additional information, such as other knowledge, hierarchical structures, and legal events, which we refer to as its "meta information". To represent these associations comprehensively, we construct a heterogeneous graph as illustrated in Fig. 3.

Specifically, the heterogeneous graph $\mathcal{G} = (N, E)$ contains different types of nodes and edges. Firstly, we categorize the nodes into three types: (i) Knowledge nodes, $N_c = \{n_i^c\}$ represent individual pieces of legal knowledge. (ii) Hierarchy nodes, $N_h = \{n_i^h\}$ represent the hierarchical structure of knowledge within the tree-like Chinese legal system (Qin et al., 2024) (e.g., "Article 273" belongs to "criminal law/property infringement/theft"). (iii) Legal event nodes, $N_e = \{n_i^e\}$, represent the legal

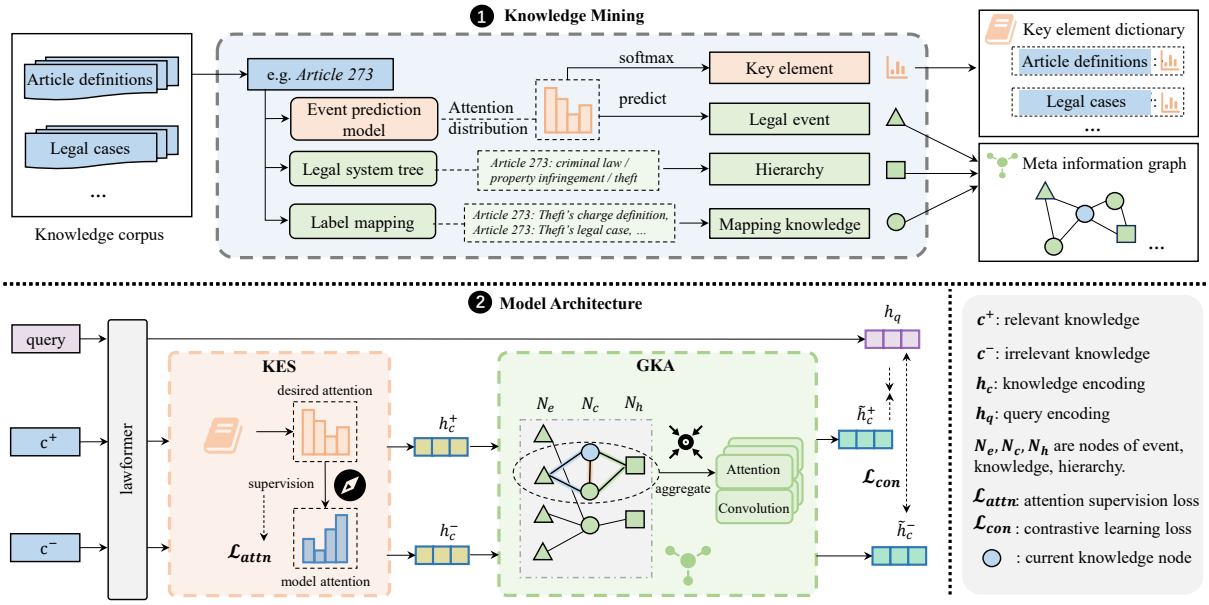

Figure 2: The overview of our UniLR. (1) is the knowledge mining process. We leverage the expertise of a legal event prediction model (Yao et al., 2022) to extract key elements into a dictionary and explore meta information to construct a heterogeneous graph. (2) is the model architecture. When encoding knowledge, KES guides the model to focus on key elements through attention supervision. GKA aggregates meta information to further enrich knowledge representation.

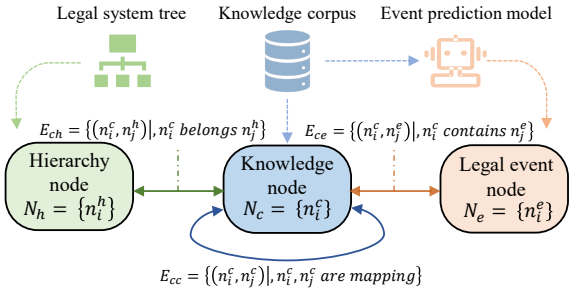

Figure 3: The definition of nodes and edges.

event labels predicted by DMBERT. We use the event label definitions from LEVEN. For node feature initialization, we use Lawformer (Xiao et al., 2021), a BERT variant pre-trained on extensive legal texts to obtain embedding.

The edges in the graph are defined based on specific connection rules: (i) If a knowledge node n_i^c belongs to a hierarchy, whose node is n_j^h, an edge is created and belongs to E_{ch}. (ii) If a knowledge node n_i^c contains the legal event, whose node is n_j^e, an edge is created and belongs to E_{ce}. (iii) If two knowledge nodes n_i^c and n_j^c have the same charge label, we think they are mapping, an edge is created and belongs to E_{cc}.

3.3 Model Architecture

3.3.1 Key Element Supervisor

To guide the retriever to focus on key elements when encoding knowledge, we introduce an attention supervision method. For the knowledge c, we input it into Lawformer. According to Eq. 2, we extract the model's attention distribution of each token and obtain $P = \{p_1, ..., p_{l_c}\}$. Next, we refer to the key element dictionary $\mathcal{D}$ to obtain the desired attention distribution $\hat{P}$ corresponding to knowledge c. To align the model's attention with the desired distribution, we introduce a loss function $\mathcal{L}_{attn}$, which minimizes the Kullback-Leibler (KL) divergence between the two distributions:

$$\mathcal{L}_{attn} = \sum_{i=1}^{l_c} KL(\hat{P}_i \parallel P_i) \quad (3)$$

Here, l_c denotes the number of tokens in the knowledge. Then we obtain the hidden state of the final layer by applying the aligned attention distribution, denoted as $h^c = [h_1; ...; h_{l_c}] \in \mathbb{R}^{l_c \times d}$, where d is the hidden dimension. By employing this approach, we effectively distill the expertise of DMBERT into the retriever, guiding the model to focus on key legal elements rather than the format.

3.3.2 Graph-based Knowledge Augmenter

To further enrich knowledge representation, we design a graph-based approach to aggregate meta

information. First, we extract the subgraph relevant to the knowledge from the entire graph $\mathcal{G}$. Specifically, for a piece of knowledge c, we identify the neighboring nodes to form a node set N', which consists of n' nodes. The edges connecting these nodes form an edge set E'. From this, we obtain a subgraph $\mathcal{G}' = \{N', E'\}$ relevant to the current knowledge.

Inspired by Guo et al. (2019), we employ graph attention to facilitate node interactions. Given GKA comprising L layers, the node representation at the l-th layer is denoted as $H^{(l)} = [h_1^{(l)}; ...; h_{n'}^{(l)}]$. To ensure that a node is not influenced by nodes that are not directly connected, we prepare a mask for each node, which defines the index set I_i of connected nodes for the i-th node. We then apply the Multi-Head Attention mechanism to learn multiple sets of attention weights, with the attention matrix for the m-th head given as follows:

$$I_i = \{j \mid i, j \in N', (n_i \leftrightarrow n_j)\} \tag{4}$$

$$\tilde{A}^{(m)} = softmax(\frac{(h_i^{(l)} W_i^Q) \cdot (h_i^{(l)} W_i^K)^T}{\sqrt{d}}), i \in I_i \tag{5}$$

Here, $W_i^Q \in \mathbb{R}^{d \times d}$ and $W_i^K \in \mathbb{R}^{d \times d}$ are learnable weight matrices used for linear transformations, $\leftrightarrow$ denotes existing edge between nodes.

To capture the high-order interactions, we design the heterogeneous graph convolution that considers different edge types. For the i-th node, we concatenate its encoding with the output from previous layers to serve as the input for the next graph layer, $g_i^{(l)} = [h^c; h_i^{(1)}; ...; h_i^{(l)}]$. The convolutional operation is then performed as follows:

$$h_i^{(l+1)} = \|_{m=1}^{M} \sigma(\sum_{j=1}^{n'} \tilde{A}_{ij}^{(k)} \cdot W_m^{(l)} \cdot g_j^{(l)} \cdot t_{ij} + b_m^{(l)}) \tag{6}$$

Here, M is the number of attention heads, $W_m^{(l)}$ is learnable weights for node features, t_{ij} is a one-hot vector indicating the edge type between n_i and n_j, and σ is the activation function. After propagating through the graph network, we aggregate the node features in the final layer:

$$H^{(L)} = MaxPooling[h_1^{(L)}; ...; h_{n'}^{(L)}] \in \mathbb{R}^{T' \times d} \tag{7}$$

To ensure stable gradient propagation, we flatten the node feature to $H_{flat}^{(L)} \in \mathbb{R}^{T'd}$, then concatenate it with the initial feature h^c and reduce the dimension via a projection layer:

$$\tilde{h}^c = RELU(W \cdot [H_{flat}^{(L)}; h^c] + b) \in \mathbb{R}^d \tag{8}$$

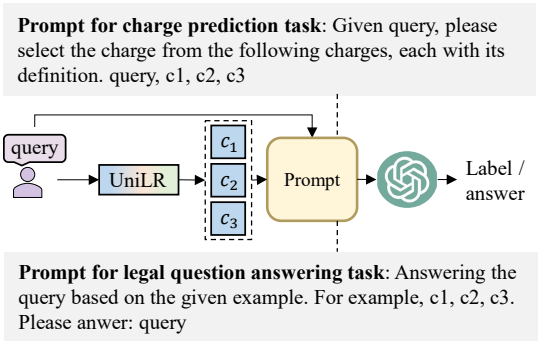

Figure 4: The process of performing legal tasks using UniLR and LLM in a RAG framework.

$W \in \mathbb{R}^{(T'd+d) \times d}$ is the weight matrix, and $b \in \mathbb{R}^{d \times 1}$ is the bias vector.

3.4 Contrastive Learning

Inspired by Gao et al. (2021), we use contrastive learning to concurrently train our model across multiple legal retrieval tasks to achieve uniformity.

For each query q, we obtain positive samples Pos by selecting knowledge with the same charge label. For example, for a *Theft* case, we choose the charge definitions of *Theft*, *Article 273*, and other cases involving *Theft*. Negative samples Neg are obtained through random sampling and hard negative sampling. When random sampling, we randomly select knowledge with different charge labels. When hard negative sampling, we select knowledge closely related within the same legal hierarchy, providing more challenging contrasts. Then, we permute positive and negative knowledge pairs and train our model by the following loss:

$$\mathcal{L}_{con} = -\sum_{\tilde{h}_c^+ \in Pos} log \frac{exp(sim(h_q, \tilde{h}_c^+)/\tau)}{\sum_{\tilde{h}_c^- \in Neg} exp(sim(h_q, \tilde{h}_c^-)/\tau)} \tag{9}$$

$h_q, \tilde{h}_c^+, \tilde{h}_c^-$ are representations of query, positive samples, and negative samples from UniLR, sim calculates the cosine similarity, and τ is the temperature that adjusts the contrastive strength.

3.5 Training and Inference

In the training process, we combine data from multiple tasks for joint training. We employ contrastive learning loss to bring the relevant knowledge representation closer to the query, and attention distribution loss to emphasize the key elements. Therefore, the overall training objective of UniLR is to minimize the following combined loss:

$$\mathcal{L}_{total} = \mathcal{L}_{con} + \lambda \mathcal{L}_{attn} \tag{10}$$

Types	Article	Charge	CVG	QA
# Train	70973	70973	97863	8451
# Test	1683	1683	2039	387
# Knowledge corpus	483	452	6124	1248
Avg. # query length	440	440	558	45
Avg. # knowledge length	113	160	571	401

Table 1: Dataset details and knowledge base.

In the inference process, we employ UniLR to encode query and knowledge. Then, the cosine similarity is used to compute their relevance scores. The knowledge with the highest relevance scores is selected to assist the LLM in solving legal tasks. As shown in Fig. 4, for the charge prediction task, UniLR retrieves several label definitions and LLM selects the final label from them. A similar process is used for the law article prediction task. For the legal question answering task, we retrieve similar QA pairs and employ few-shot learning to guide LLM in producing the final answer. A similar process is used for the court view generation task.

4 Experiments

4.1 Dataset details and knowledge base

We conduct experiments on multiple common legal tasks: law article prediction (Article), charge prediction (Charge), court view generation (CVG), and legal question answering (QA). The dataset details and corresponding knowledge base are shown in Tab.1. **CAIL2018-Article** and **CAIL2018-Charge** are from the Chinese AI and Law challenge legal judgment prediction dataset [3] (Xiao et al., 2018). To align with prior works (Xu et al., 2020; Yue et al., 2021a), we filter out data with multiple articles and charges. In these datasets, the query is a fact description, and the task is to predict article and charge labels, respectively. **LAIC2021-CVG** is the court view generation dataset from Legal AI challenge [4], where the query is a fact description and the task is to generate the court view. **Lawzhidao-QA** is the criminal question-answering dataset selected from Baidu Legal Question Answering competition[5], where the query is a legal question and the task is to generate the answer. The details of the knowledge source can be found in Appendix A.

[3]http://cail.cipsc.org.cn/index.html
[4]https://data.court.gov.cn/pages/laic2021.html
[5]https://aistudio.baidu.com/datasetdetail/95693

4.2 Baselines

We implement three parts of baselines for comprehensive comparison and set retrieved knowledge number $k = 3$, with experimental settings detailed in Appendix B. We also conduct further experiments in Appendix C, including performance variation with different k values and pre-trained models and runtime analysis.

Traditional task-specific methods. For prediction tasks, **TopJudge** (Zhong et al., 2018) captures topological dependencies among the subtasks in legal judgment prediction. **LADAN** (Xu et al., 2020) uses graph distillation for distinguishing charges and law articles. **NeurJudge** (Yue et al., 2021a) splits fact descriptions using intermediate subtask results for prediction. For generation tasks, **BART** (Lewis et al., 2019) is a widely used bidirectional autoregressive Transformer model. **T5** (Raffel et al., 2020) is a transformer architecture model that follows a text-to-text transfer learning paradigm. **C3VG** (Yue et al., 2021b) follows a two-stage architecture which is extraction-generation.

LLM methods. For LLMs, we use **GPT4** (Achiam et al., 2023), **LLaMA-3** (Cui et al., 2023b), **GLM4** (Zeng et al., 2024). For legal LLMs, we use **LexiLaw** [6], **LaWGPT** (Zhou et al., 2024b).

LLM with retriever methods. To evaluate retrieval enhancements, we implement the following retrieval methods with top-performing LLMs: **BM25** (Robertson and Zaragoza, 2009) is a retrieval model based on term frequency and document length. **Contriever** (Izacard et al., 2022) is a dense embedding model trained via contrastive learning. **LED** (Zhang et al., 2023) enhances dense retrieval by aligning embeddings with lexicon-aware representations through weakened knowledge distillation. **SAILER** (Li et al., 2023a) is a legal case retriever that incorporates structural information and legal rules.

4.3 Evaluation Metrics

4.3.1 Retrieval evaluation

We use Hit@k (Norouzi et al., 2014) as the evaluation metric. if there is relevant knowledge among the k knowledge retrieved, it is considered successful retrieval. Considering that the relevant knowledge is not unique when retrieving precedents, we

[6]https://github.com/CSHaitao/LexiLaw

Methods	CAIL2018-Article					CAIL2018-Charge				
	Ma-P	Ma-R	Ma-F	Acc	Hit@k	Ma-P	Ma-R	Ma-F	Acc	Hit@k
Traditional task-specific methods										
TopJudge	74.49	66.26	68.68	80.85	-	74.43	68.37	70.41	78.42	-
LADAN	75.61	70.29	70.46	80.92	-	75.36	70.02	71.19	79.45	-
NeurJudge	75.16	72.01	72.26	**81.94**	-	75.76	71.24	71.59	80.31	-
LLM methods										
GPT4	14.76	13.46	13.47	14.68	-	45.97	35.61	34.38	46.94	-
LLaMA-3	15.18	12.43	13.19	14.00	-	40.51	31.88	33.09	42.00	-
GLM4	9.42	7.47	6.57	14.00	-	47.17	35.27	37.68	41.33	-
LexiLaw	12.27	6.54	7.41	9.33	-	36.59	32.81	37.73	42.33	-
LaWGPT	14.49	12.17	12.64	16.35	-	32.21	28.81	34.11	43.37	-
Best LLM (GPT4) with retriever methods										
+ BM25	31.90	25.58	26.70	36.82	71.13	52.63	42.87	46.17	50.75	73.28
+ Contriever	72.35	63.41	67.83	73.76	75.06	73.10	65.90	66.09	71.37	81.58
+ LED	70.73	63.24	67.97	74.54	78.31	74.41	66.79	68.42	69.62	82.25
+ SAILER	73.75	65.85	69.76	77.91	80.70	74.76	68.95	69.70	76.75	83.90
+ UniLR	**77.24**	**72.73**	**72.32**	80.98	**84.88**	**76.98**	**73.11**	**72.34**	**80.36**	**85.27**

Table 2: The performance of article prediction task and charge prediction task. The best is **bolded**, the second best is underlined, and the gray-shaded metric only measures retriever performance when $k = 3$.

Methods	LAIC2021-CVG						Lawzhidao-QA					
	B-1	B-2	B-N	R-L	Hit@k	R-p@k	B-1	B-2	B-N	R-L	Hit@k	R-p@k
Traditional task-specific methods												
Bart	61.01	52.01	48.97	56.95	-	-	37.58	23.71	20.74	21.93	-	-
T5	61.24	51.24	47.68	58.09	-	-	36.32	21.45	20.24	18.31	-	-
C3VG	63.35	52.70	49.30	60.71	-	-	-	-	-	-	-	-
LLM methods												
GPT4	41.23	26.90	23.06	27.46	-	-	11.98	7.63	5.73	14.38	-	-
LLaMA-3	35.98	19.74	18.01	23.17	-	-	20.70	15.27	14.10	18.39	-	-
GLM4	42.15	28.01	22.32	32.31	-	-	33.02	20.10	15.43	20.73	-	-
LexiLaw	39.30	23.26	20.59	23.91	-	-	28.99	16.56	14.42	19.23	-	-
LaWGPT	15.17	10.03	8.59	12.85	-	-	24.32	16.10	14.22	17.78	-	-
Best LLM (GLM4) with retriever methods												
+ BM25	53.47	46.53	44.46	58.30	83.20	65.43	39.04	28.24	24.49	30.72	59.63	37.47
+ Contriever	57.35	51.07	48.98	63.51	87.50	77.32	42.26	30.38	26.42	32.95	65.22	55.10
+ LED	57.49	51.78	48.84	63.52	87.88	79.46	43.80	32.45	27.54	32.87	66.78	55.73
+ SAILER	59.06	53.02	50.98	65.05	90.97	87.86	45.00	34.93	30.82	36.26	70.43	60.73
+ UniLR	**63.94**	**58.03**	**55.81**	**72.68**	**91.41**	**90.23**	**48.38**	**37.13**	**33.27**	**40.87**	**75.90**	**64.10**

Table 3: The performance of court view generation task and legal question answering task. The best is **bolded**, the second best is underlined, and the gray-shaded metric measures retriever performance when $k = 3$.

select R-p@k (Chen et al., 2023) to do further evaluation. It is calculated as r/R, where R is the number of retrieved knowledge and r is the number of relevant knowledge.

4.3.2 Task performance evaluation

For the prediction tasks, we employ macro precision (Ma-P), macro recall (Ma-R), macro f1 score (Ma-F), and accuracy (Acc). For the generation task, we use BLEU-1 (B-1), BLEU-2 (B-2), BLEU-N (B-N, the average value of BLEU-1 $\sim$4), and ROUGE-L (R-L) (Lin, 2004). To comprehensively test the generation results, we conduct the human evaluation, with details provided in Appendix D.

4.4 Experimental Results

4.4.1 The performance of prediction tasks

From Tab.2, we can conclude that: (1) Traditional task-specific methods excel over LLMs. This is because LLMs are generative, selecting a token from a vast vocabulary in decoding. (2) Legal LLMs underperform universal LLMs like GPT4, possibly due to a decrease in in-context learning ability during fine-tuning. (3) In retrieval evaluation, our UniLR outperforms the best baseline SAILER by 5.18% in Hit@k for article prediction and 1.63% for charge prediction. This indicates the effectiveness of introducing key elements and meta information. (4)In prediction evaluation, UniLR significantly improves GPT4's perfor-

Methods	CAIL2018-Article					CAIL2018-Charge				
	Ma-P	Ma-R	Ma-F	Acc	Hit@k	Ma-P	Ma-R	Ma-F	Acc	Hit@k
UniLR	**77.24**	**72.73**	**72.32**	**80.98**	**84.88**	**76.98**	**73.11**	**72.34**	**80.36**	**85.27**
w/o KES	74.15	71.31	70.76	78.87	81.33	75.25	69.02	69.83	77.89	81.68
w/o GKA	75.70	71.80	71.04	79.57	81.96	74.18	71.07	68.59	78.69	82.37
Ablation of meta information in GKA										
w/o E_{cc}	75.41	71.28	70.91	79.09	82.94	75.97	70.24	70.07	78.53	83.01
w/o E_{ce}	76.45	72.64	72.07	80.57	84.14	76.42	72.26	71.83	79.59	84.58
w/o E_{ch}	75.10	71.65	71.06	80.12	84.05	76.15	69.48	70.30	77.39	82.94

Table 4: Ablation experiment of the best performing LLM with UniLR in prediction tasks.

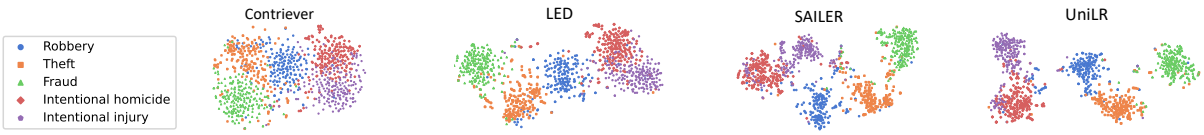

Figure 5: The t-SNE plots of legal cases.

Methods	CVG		QA	
	Flu.	Rat.	Flu.	Rat.
BART	4.44	3.64	2.18	2.52
GLM4	3.37	2.71	3.09	2.91
GLM4+SAILER	4.46	3.93	3.95	3.47
GLM4+UniLR	**4.62**	**4.04**	**4.28**	**3.99**

Table 5: Human evaluation on legal generation tasks.

mance, surpassing all baselines, including the traditional SOTA method. This suggests that UniLR effectively bridges the gap between LLMs and legal tasks.

4.4.2 The performance of generation tasks

From Tab.3 and Tab.5, we have the following observations: (1) LLM methods show poor performance, indicating their tendency to generate imaginative outputs are unsuitable for the precision required in the legal domain. (2) Many LLM with retriever methods surpass traditional task-specific methods emphasizing the importance of knowledge. (3) GLM4 surpasses GPT4, suggesting that its training involved more Chinese legal documents. (4) UniLR achieves top retrieval performance and significantly boosts GLM4's downstream task performance. For example, in legal QA, it outperforms the second-best baseline by 7.77% in Hit@k and 12.71% in R-L. (5) In human evaluation, UniLR excels in Fluency and Rationality, aligning with automatic evaluations. Additionally, BART performs well in CVG but lags in QA, while GLM4 shows the opposite trend due to the structured nature of court views versus the flexibility of QA tasks.

4.4.3 Ablation study

We conduct ablation experiments on the GPT4+UniLR method in two prediction tasks, as shown in Tab. 4. (1) **w/o KES** eliminates the key elements supervision, relying solely on Lawformer encoding. We find a significant performance degradation, demonstrating the importance of focusing on key elements. (2) **w/o GKA** removes the graph-based knowledge augmenter. Noticeable performance degradation indicates that aggregating meta information can effectively capture a comprehensive knowledge representation. (3) We conduct ablation experiments on GKA's associations, including mapping knowledge, legal events, and hierarchy. Removing associations between knowledge results in the most significant performance decline.

4.5 Visual Analysis

Following Li et al. (2023a), to explore UniLR's distinguish ability for legal knowledge representation, we select 5000 illegal facts from CAIL2018 that involve confusing charges and visualize their encoding by using different retriever methods. Specifically, the charges included *Robbery, Theft, Snatch, Intentional injury, Intentional homicide*, each with 1000 cases. As observed from Fig. 5, UniLR significantly increases the distance between encoded illegal facts related to different charges, demonstrating its strongest ability to distinguish between confusing knowledge. Case Study in Appendix E also confirms the effectiveness of UniLR.

5 Conclusion and Future Work

In conclusion, we address the challenges in enhancing LLMs in multiple legal tasks by introducing a Unified Legal Knowledge Retriever (UniLR). By incorporating key elements and meta information, UniLR significantly alleviates disparities in multiple retrieval tasks and reduces knowledge confusion. Extensive experiments on multiple common legal tasks demonstrate that UniLR outperforms state-of-the-art retrieval methods and significantly enhances the performance of LLMs in legal applications. In the future, to better explore Retrieval-Augmented Generation (RAG) in legal tasks, we aim to make efforts in two directions: (1) Training an LLM capable of flexibly utilizing knowledge obtained from retrievers. (2) Leveraging feedback from the LLM to train a retriever that is better suited for LLM.

6 Ethical Issue Discussion

Legal AI has benefited from the emergence of LLMs, but it's a sensitive technology that demands ethical considerations. Our UniLR is designed to enhance LLMs in legal tasks, mitigating the risk of factual errors to some extent. However, even minor inaccuracies could have significant consequences. Our goal is to provide suggestions to judges rather than replace them. In practice, human judges should be the final safeguard to protect justice and fairness. Although our method demonstrates promising results on legal task datasets, it does not imply that it can endow LLMs with human-like empathy, experience, and intuition. It is necessary to prevent misuse. Additionally, given that the model retrieves knowledge from the external corpus, ensuring the quality and fairness of the corpus is crucial.

7 Limitations

In this section, we discuss the limitations of our works as follows:

- We do not pretrain or fine-tune LLMs for the RAG process. Training an LLM capable of flexibly utilizing knowledge obtained from retrievers, may produce better output.
- We validate the effectiveness of designing a legal retriever to meet LLM needs. Exploring the application of such retrievers in other fields like medicine and education is worth considering.
- Our research is conducted on the Chinese legal system. We are also very interested in exploring the generalization of our methods to other languages.

8 Acknowledgments

This work was supported in part by "Pioneer" and "Leading Goose" R&D Program of Zhejiang (2025C02037), National Natural Science Foundation of China (62376243, 62441605), and National Key Research and Development Program of China (2024YFE0203700). All opinions in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Rose Biderman. 2021. Gpt-neo: Large scale autoregressive language modeling with mesh-tensorflow.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

S.J. Burton. 2007. *An Introduction to Law and Legal Reasoning*. Academic Success Series. Walters Kluwer Law & Business.

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

Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yiqun Liu, Yixing Fan, and Xueqi Cheng. 2023. A unified generative retriever for knowledge-intensive language tasks via prompt learning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23. ACM.

Yen-Liang Chen, Yi-Hung Liu, and Wu-Liang Ho. 2013. A text mining approach to assist the general public in the retrieval of legal documents. *Journal of the American Society for Information Science and Technology*, 64(2):280–290.

Jiaxi Cui, Zongjian Li, Yang Yan, Bohua Chen, and Li Yuan. 2023a. Chatlaw: Open-source legal large language model with integrated external knowledge bases. *Preprint*, arXiv:2306.16092.

Yiming Cui, Ziqing Yang, and Xin Yao. 2023b. Efficient and effective text encoding for chinese llama and alpaca. *arXiv preprint arXiv:2304.08177*.

Zhiwei Fei, Xiaoyu Shen, Dawei Zhu, Fengzhe Zhou, Zhuo Han, Alan Huang, Songyang Zhang, Kai Chen, Zhixin Yin, Zongwen Shen, et al. 2024. Lawbench: Benchmarking legal knowledge of large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7933–7962.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Suyu Ge, Chenyan Xiong, Corby Rosset, Arnold Overwijk, Jiawei Han, and Paul Bennett. 2023. Augmenting zero-shot dense retrievers with plug-in mixture-of-memories. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1796–1812.

Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention guided graph convolutional networks for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 241–251.

Quzhe Huang, Mingxu Tao, Zhenwei An, Chen Zhang, Cong Jiang, Zhibin Chen, Zirui Wu, and Yansong Feng. 2023. Lawyer llama technical report. *ArXiv*, abs/2305.15062.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Preprint*, arXiv:2112.09118.

Phi Manh Kien, Ha-Thanh Nguyen, Ngo Xuan Bach, Vu Tran, Minh Le Nguyen, and Tu Minh Phuong. 2020. Answering legal questions by learning neural attentive text representation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 988–998, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Ang Li, Qiangchao Chen, Yiquan Wu, Ming Cai, Xiang Zhou, Fei Wu, and Kun Kuang. 2024a. From graph to word bag: Introducing domain knowledge to confusing charge prediction. *Preprint*, arXiv:2403.04369.

Ang Li, Yiquan Wu, Ming Cai, Adam Jatowt, Xiang Zhou, Weiming Lu, Changlong Sun, Fei Wu, and Kun Kuang. 2025. Legal judgment prediction based on knowledge-enhanced multi-task and multi-label text classification. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6957–6970, Albuquerque, New Mexico. Association for Computational Linguistics.

Ang Li, Yiquan Wu, Yifei Liu, Fei Wu, Ming Cai, and Kun Kuang. 2024b. Enhancing court view generation with knowledge injection and guidance. *Preprint*, arXiv:2403.04366.

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*, pages 1035–1044.

Qingquan Li, Qifan Zhang, Junjie Yao, and Yingjie Zhang. 2020. Event extraction for criminal legal text. In *2020 IEEE International Conference on Knowledge Graph (ICKG)*, pages 573–580.

Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. 2023b. Unified demonstration retriever for in-context learning. *CoRR*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Hongcheng Liu, Yusheng Liao, Yutong Meng, and Yuhao Wang. 2023. Lawgpt: Chinese legal dialogue language model. `https://github.com/LiuHC0428/LAW_GPT`.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for gpt-3?. *DeeLIO 2022*, page 100.

Yifei Liu, Yiquan Wu, Ang Li, Yating Zhang, Changlong Sun, Weiming Lu, Fei Wu, and Kun Kuang. 2024. Unleashing the power of LLMs in court view generation by stimulating internal knowledge and incorporating external knowledge. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2782–2792, Mexico City, Mexico. Association for Computational Linguistics.

Antoine Louis and Gerasimos Spanakis. 2022. A statutory article retrieval dataset in french. *Preprint*, arXiv:2108.11792.

Antoine Louis, Gijs van Dijck, and Gerasimos Spanakis. 2024. Interpretable long-form legal question answering with retrieval-augmented large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22266–22275.

Man Luo, Xin Xu, Zhuyun Dai, Panupong Pasupat, Seyed Mehran Kazemi, Chitta Baral, Vaiva Imbrasaite, and Vincent Y Zhao. 2023. Dr. icl: Demonstration-retrieved in-context learning. *CoRR*.

Zheqi Lv, Tianyu Zhan, Wenjie Wang, Xinyu Lin, Shengyu Zhang, Wenqiao Zhang, Jiwei Li, Kun Kuang, and Fei Wu. 2025. Collaboration of large language models and small recommendation models for device-cloud recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, KDD '25, page 962–973, New York, NY, USA. Association for Computing Machinery.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *Preprint*, arXiv:2305.14251.

Mohammad Norouzi, Tomas Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg S. Corrado, and Jeffrey Dean. 2014. Zero-shot learning by convex combination of semantic embeddings. *Preprint*, arXiv:1312.5650.

Weicong Qin, Zelin Cao, Weijie Yu, Zihua Si, Sirui Chen, and Jun Xu. 2024. Explicitly integrating judgment prediction with legal document retrieval: A law-guided generative approach. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 2210–2220, New York, NY, USA. Association for Computing Machinery.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

K. Raghav, Krishna Reddy, and V. Balakista Reddy. 2016. Analyzing the extraction of relevant legal judgments using paragraph-level and citation information.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. *Preprint*, arXiv:2112.08633.

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 *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3501–3507. International Joint Conferences on Artificial Intelligence Organization. Main track.

Shirong Shen, Guilin Qi, Zhen Li, Sheng Bi, and Lusheng Wang. 2020. Hierarchical Chinese legal event extraction via pedal attention mechanism. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 100–113, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Boxin Wang, Wei Ping, Peng Xu, Lawrence McAfee, Zihan Liu, Mohammad Shoeybi, Yi Dong, Oleksii Kuchaiev, Bo Li, Chaowei Xiao, Anima Anandkumar, and Bryan Catanzaro. 2023. Shall we pretrain autoregressive language models with retrieval? a comprehensive study. *Preprint*, arXiv:2304.06762.

Pengfei Wang, Yu Fan, Shuzi Niu, Ze Yang, Yongfeng Zhang, and Jiafeng Guo. 2019a. Hierarchical matching network for crime classification. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 325–334. ACM.

Pengfei Wang, Ze Yang, Shuzi Niu, Yongfeng Zhang, Lei Zhang, and Shaozhang Niu. 2018. Modeling dynamic pairwise attention for crime classification over legal articles. In *The 41st International ACM*

SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018, pages 485–494. ACM.

Xiaozhi Wang, Xu Han, Zhiyuan Liu, Maosong Sun, and Peng Li. 2019b. Adversarial training for weakly supervised event detection. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 998–1008, Minneapolis, Minnesota. Association for Computational Linguistics.

Yiquan Wu, Kun Kuang, Yating Zhang, Xiaozhong Liu, Changlong Sun, Jun Xiao, Yueting Zhuang, Luo Si, and Fei Wu. 2020. De-biased court's view generation with causality. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 763–780. Association for Computational Linguistics.

Yiquan Wu, Yifei Liu, Weiming Lu, Yating Zhang, Jun Feng, Changlong Sun, Fei Wu, and Kun Kuang. 2022. Towards interactivity and interpretability: A rationale-based legal judgment prediction framework. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4787–4799.

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, and Jianfeng Xu. 2018. Cail2018: A large-scale legal dataset for judgment prediction. *Preprint*, arXiv:1807.02478.

Nuo Xu, Pinghui Wang, Long Chen, Li Pan, Xiaoyan Wang, and Junzhou Zhao. 2020. Distinguish confusing law articles for legal judgment prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3086–3095, Online. Association for Computational Linguistics.

Wenmian Yang, Weijia Jia, Xiaojie Zhou, and Yutao Luo. 2019. Legal judgment prediction via multi-perspective bi-feedback network. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization.

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.

Linan Yue, Qi Liu, Binbin Jin, Han Wu, Kai Zhang, Yanqing An, Mingyue Cheng, Biao Yin, and Dayong Wu. 2021a. Neurjudge: A circumstance-aware neural framework for legal judgment prediction. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 973–982.

Linan Yue, Qi Liu*, Han Wu, Yanqing An, Li Wang, and Senchao Yuan. 2021b. Circumstances enhanced criminal court view generation. The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval.

Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, et al. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *CoRR*.

Kai Zhang, Chongyang Tao, Tao Shen, Can Xu, Xiubo Geng, Binxing Jiao, and Daxin Jiang. 2023. Led: Lexicon-enlightened dense retriever for large-scale retrieval. In *Proceedings of the ACM Web Conference 2023*, pages 3203–3213.

Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. 2018. Legal judgment prediction via topological learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3540–3549, Brussels, Belgium. Association for Computational Linguistics.

Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020a. How does NLP benefit legal system: A summary of legal artificial intelligence. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5218–5230, Online. Association for Computational Linguistics.

Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020b. Jec-qa: a legal-domain question answering dataset. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9701–9708.

Xiang Zhou, Yudong Wu, Ang Li, Ming Cai, Yiquan Wu, and Kun Kuang. 2024a. Unlocking authentic judicial reasoning: A template-based legal information generation framework for judicial views. *Knowledge-Based Systems*, 301:112232.

Zhi Zhou, Jiang-Xin Shi, Peng-Xiao Song, Xiaowen Yang, Yi-Xuan Jin, Lan-Zhe Guo, and Yu-Feng Li. 2024b. Lawgpt: A chinese legal knowledge-enhanced large language model. *CoRR*.

A Knowledge Source.

We collect multiple legal knowledge sources to construct the knowledge base. In our UniLR, they are further processed together into a legal essence dictionary and a heterogeneous knowledge metagraph.

(1) **Article definitions**: are detailed content about all criminal law articles. It assists in the article prediction task. (2) **Charge definitions**: are specific definitions for all criminal charges from the Criminal Law, formatted similarly to Xu et al. (2020). It assists in the charge prediction task. (3) **CVG cases**: are selected from the training set of LAIC2021 in a balanced way. It contains fact descriptions and corresponding court views. It assists in the court view generation task. (4) **QA cases**: are similarly selected from the training set of Lawzhidao-QA dataset. It assists in the legal QA task. (5) **Legal system tree**: is extracted from the Chinese Criminal Law, which has a tree-like structure comprising chapters and specific offenses. Each knowledge can be brought into a leaf node in the hierarchical structure, utilizing paths from the root to the leaf. (6) **Legal events**: are from the legal event detection dataset LEVEN (Yao et al., 2022). We focus on charge-oriented events and leverage their definitions.

B Experiment Settings

All training and inference were conducted on 2 NVIDIA Tesla A100 GPUs. We rerun the experiments five times with different random seeds and report the average. We also use the Fisher randomization test to ensure the significance of the results.

B.1 Retrieval setting

We set the maximum length for input query and each knowledge to 512. For the BM25 algorithm, we set $k1$ to 1.5 and b to 0.75. Other retrieval methods utilizing pre-trained models (Contriever, LED, SAILER, UniLR) leverage the Lawformer (Xiao et al., 2021) as the pre-trained embedding model, followed by fine-tuning. To ensure fairness, the dense retrieval baselines are jointly fine-tuned on multiple legal retrieval tasks, similar to UniLR. In the fine-tuning, all retrievers are trained using the Adam optimizer (Kingma and Ba, 2017) with a learning rate of 1e-6 for 50 epochs. In our UniLR, we set the GKA layers to 2, and the hyperparameter controlling loss λ to the best-performing value of 0.2. In the reference of retrieval, we use top-k selection to select the final knowledge and set k=3. Referring to Langhain Chatchat [7], we use FAISS to pre-vectorize the knowledge base, allowing for quick computation of relevance scores with the query during inference.

[7]https://github.com/chatchat-space/Langchain-Chatchat

B.2 Legal tasks setting

For traditional methods, they are trained and inferred according to their original papers. For all LLMs, we set the inference temperature to 0.8 and the maximum output length to 512. Regarding LLM with retrieval methods, since they require concatenating the query and knowledge, we set the maximum input length to 2048.

C Further Experiments

C.1 The performance variation with different k values

In this paper, we set the number of retrieved knowledge number $k = 3$ for each query. We explore the impact of different k values on the performance of multiple legal retrieval tasks, testing experimental results for k ranging from 1 to 5. As shown Tab. 6, when $k \leq 3$, the performance gain from adding retrieved knowledge number is relatively significant. However, further increasing k brings minimal improvement. Moreover, inference with the LLM consumes substantial computational resources. Considering the trade-off between performance gain and resource consumption, we chose $k = 3$ for the main experiment.

C.2 Sensitivity analysis of the combined loss hyperparameter

To evaluate the effect of the hyperparameter λ in the combined loss function, we conducted experiments on four datasets by varying λ within the range [0.1,0.4]. The results are shown in Tab. 7. The results indicate that when $\lambda < 0.2$, the attention supervision signal is insufficient, resulting in lower performance. Conversely, when $\lambda > 0.2$, the attention supervision component dominates the optimization objective and suppresses the primary task loss, leading to degraded performance. Optimal results are achieved when $\lambda = 0.2$, which provides a balanced trade-off between task-specific loss and attention supervision.

C.3 The performance variation with different pre-trained model

To assess the robustness of our pipeline, we replaced Lawformer with BERT-base Chinese. We also set the retrieved knowledge number $k = 3$. The experimental results for law article prediction task and charge prediction task are shown in Tab. 8 and Tab. 9. The BERT pre-trained model was not specifically trained on legal texts, yet it did

k	CAIL2018-Article	CAIL2018-Charge	LAIC2021-CVG	Lawzhidao-QA
1	75.72	76.37	90.63	65.96
2	80.73	81.91	91.21	72.17
3	84.88	85.27	91.41	75.90
4	86.05	86.25	91.60	77.76
5	86.84	87.22	91.99	78.39

Table 6: Performance of Different k Values

λ	CAIL2018-Article	CAIL2018-Charge	LAIC2021-CVG	Lawzhidao-QA
0.1	83.78	84.11	90.41	73.93
0.2	**84.88**	**85.27**	**91.41**	**75.90**
0.3	84.71	83.59	91.54	74.11
0.4	84.40	82.49	90.78	73.20

Table 7: Performance with different values of λ in the combined loss function.

Methods	Ma-F	Acc	Hit@k
UniLR (BERT)	71.84	80.73	82.84
UniLR (Lawformer)	72.32	80.98	84.88

Table 8: The performance of law article prediction task.

Methods	Ma-F	Acc	Hit@k
UniLR (BERT)	72.22	79.89	84.96
UniLR (Lawformer)	72.34	80.36	85.27

Table 9: The performance of charge prediction task.

not cause a significant performance drop in UniLR. These results demonstrate that UniLR maintains robust performance when replacing the pre-trained model.

C.4 Runtime analysis

We further analyze the runtime as follows: During training, we used a distributed setup on a single machine with 2 A100 GPUs. Each GPU utilized 16,964MB of memory, with a batch size of 8. The model was trained for 50 epochs, completing in approximately 5.56 hours. For inference, the model uses FAISS to pre-store knowledge as a vector database, which is 44.4MB in size. During inference, the model occupies 4,356MB of GPU memory. Retrieving results from the model is quick, with the primary time consumption being LLMs generating answers based on the retrieved knowledge. On average, generating each response takes 1.53 seconds.

D Human Evaluation Metric

We select baselines with good similarity performance to do human evaluation. We randomly select 200 samples and shuffle them to ensure fairness. We invite five annotators (10 Ph.D. students from the Law major) to evaluate every sample from two perspectives referencing true labels:

- **Fluency.** The annotators rate the fluency of generation texts on a scale of 1-5.
- **Rationality.** The annotator needs to score 1-5 on whether the answer to the question is reasonable.

When scoring on a scale of 1-5, they are required to provide integer scores, i.e., selecting from the range [1, 2, 3, 4, 5].

E Case Study

We conduct the case study to demonstrate the application details of UniLR in legal prediction tasks. In Fig. 6, based on the fact description, the defendant was discovered stealing property and then engaged in violent behavior (forcibly dragging the victim), which constituted a robbery. UniLR correctly identified this and accurately predicted *Robbery*, while NeurJudge predicted theft based on the act of *Theft*.

We also provided a case for legal QA in Fig. 7, we use gray highlights to represent confusing semantics. Red highlights represent key elements, while green highlights represent meta information. It is observed that SAILER, influenced by the format *in order to* and the word *detain*, retrieves a *Detention* case. In reality, due to the defendant's intention for valuables and the violent actions, the charge has transformed into *Kidnapping*. UniLR focuses on key elements such as valuables and violence in the knowledge, and further enriches the knowledge information by associating the legal

Fact Description	At around 2 a.m. on August 4, 2015, the defendant, W, drove a tricycle to a foot bath shop in Cixi City and stole a white Apple 4S phone worth RMB 750 while the victim, L, was asleep. As W was escaping, L woke up and grabbed the tricycle. Despite knowing this, W accelerated, dragging L for about 20 meters and causing L to fall and sustain minor injuries. The stolen phone was recovered and returned to L.			
Judgment		Ground truth	NeurJudge ✗	UniLR ✓
	Law articles	Article 263	**Article 263** (Anyone who steals a significant amount of public or private property, commits theft multiple times, breaks into a residence to steal, carries a weapon during theft, or engages in pickpocketing shall be sentenced to up to 3 years of imprisonment, detention, or control, and fined..)	**Article 263** (Anyone who robs public or private property by violence, threat, or other means shall be sentenced to 3 to 10 years of imprisonment and fined)
	charges	Robbery	**Theft** (Definition: Theft is the act of secretly taking a significant amount of public or private property, or repeatedly secretly taking public or private property, with the intent of illegal possession.)	**Robbery** (Definition: Robbery is the act of unlawfully taking public or private property by force, threat, or other means with the intent of illegal possession.)

Figure 6: A case study of article and charge prediction tasks.

Query	In order to quickly obtain valuables, defendant A detained victim B for an extended period and violently demanded cash. Is the sentence in this case serious?	
Ground Truth	According to Article 239 of the Criminal Law, the defendant's behavior seriously violates the individual's right to freedom and personal safety. Defendant A kidnap the victim for the purpose of forcing him to provide property, and he shall be sentenced to fixed-term imprisonment of not less than ten years or life imprisonment.	
Retrieved Knowledge	SAILER ✗	UniLR ✓
	Query: In order to seek revenge, defendant A illegally detained and insulted Victim B. How will he be sentenced? **Answer:** According to the law, illegally detaining others shall be sentenced to fixed-term imprisonment of not more than three years, detention, public surveillance, or deprivation of political rights.	**Query:** Defendant A kidnapped and extorted Victim B 5 million yuan. The stolen money has been spent and A can not repay civil compensation. What is the verdict? **Answer:** According to the law, kidnapping for ransom of 5 million yuan is suspected of kidnapping and should be sentenced to more than ten years in prison or life imprisonment. **Legal event Kidnapping:** Using violent means to take hostages in exchange for benefits, the target is a person. **Article knowledge:** Article 239, Anyone who kidnaps others with the purpose of extorting money or property shall be sentenced to at least ten years…

Figure 7: A case study of Legal QA task.

event and the article. Ultimately, UniLR successfully retrieves the appropriate QA case of *Kidnapping*.