Explicitly Integrating Judgment Prediction with Legal Document Retrieval: A Law-Guided Generative Approach

Weicong Qin
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
qwc@ruc.edu.cn

Zelin Cao
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
zelin_cao@ruc.edu.cn

Weijie Yu
School of Information Technology and Management
University of International Business and Economics
Beijing, China
yu@uibe.edu.cn

Zihua Si
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
zihua_si@ruc.edu.cn

Sirui Chen
University of Illinois at Urbana-Champaign
Illinois, USA
chensr16@gmail.com

Jun Xu*
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
junxu@ruc.edu.cn

ABSTRACT

Legal document retrieval and judgment prediction are crucial tasks in intelligent legal systems. In practice, determining whether two documents share the same judgments is essential for establishing their relevance in legal retrieval. However, existing legal retrieval studies either ignore the vital role of judgment prediction or rely on implicit training objectives, expecting a proper alignment of legal documents in vector space based on their judgments. Neither approach provides explicit evidence of judgment consistency for relevance modeling, leading to inaccuracies and a lack of transparency in retrieval. To address this issue, we propose a law-guided method, namely GEAR, within the generative retrieval framework. GEAR explicitly integrates judgment prediction with legal document retrieval in a sequence-to-sequence manner. Specifically, given the intricate nature of legal documents, we first extract rationales from documents based on the definition of charges in law. We then employ these rationales as queries, ensuring efficiency and producing a shared, informative document representation for both tasks. Second, in accordance with the inherent hierarchy of law, we construct a law structure constraint tree and represent each candidate document as a hierarchical semantic ID based on this tree. This empowers GEAR to perform dual predictions for judgment and relevant documents in a single inference, i.e., traversing the tree from the root through intermediate judgment nodes, to document-specific leaf nodes. Third, we devise the revision loss that jointly minimizes the discrepancy between the IDs of predicted and labeled judgments, as well as retrieved documents, thus improving accuracy and consistency for both tasks. Extensive experiments on two Chinese legal case retrieval datasets show the superiority of GEAR over state-of-the-art methods while maintaining competitive judgment prediction performance. Moreover, we validate the effectiveness of GEAR on a French statutory article retrieval dataset, reaffirming its robustness across languages and domains.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; • **Applied computing** → *Law*.

KEYWORDS

Legal Document Retrieval, Generative Retrieval, Legal Judgment Prediction

ACM Reference Format:
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), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3626772.3657717

1 INTRODUCTION

Legal document retrieval and judgment prediction are fundamental components in intelligent legal systems. The former entails the retrieval of relevant legal documents (cases or statutory articles) give a query. On the other hand, the latter seeks to predict the outcomes or judgments rendered in legal cases, such as applicable charges, term-of-penalties, etc.

These two tasks are closely intertwined [28, 30, 45] in practice. From the retrieval side, determining whether two documents share the same judgments is essential for establishing their relevance.

*Jun Xu is the corresponding author. Work partially done at Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education.

SIGIR '24, July 14–18, 2024, Washington, DC, USA

ACM ISBN 979-8-4007-0431-4/24/07
https://doi.org/10.1145/3626772.3657717

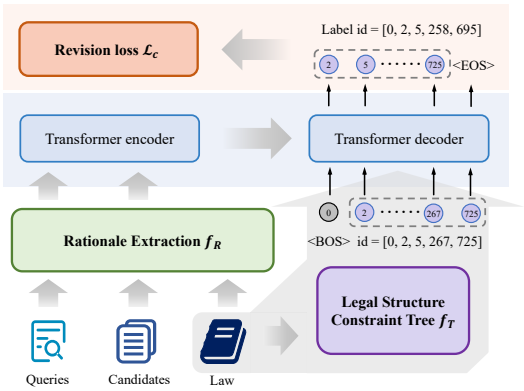


Figure 1: The overview of our proposed GEAR. It mainly consists of three modules, including rationale extraction, law structure constraint tree, and revision loss. The middle part in blue is the generative retrieval framework.

Regrettably, most of the existing studies [7, 21, 33, 37, 41] of legal document retrieval frequently overlook the significance of judgment prediction and merely focus on the text-level semantic similarity. Recently, Li et al. [15] introduced an implicit training objective that uses the fact description of the legal document to predict its judgment, expecting a proper alignment of legal documents in vector space based on their judgments. While these studies show effectiveness in retrieval performance, they fail to provide explicit evidence of judgment consistency for relevance modeling. Consequently, this limitation leads to inaccuracies and a lack of transparency [2, 12, 24]. It is because their legal relevance reasoning especially regarding judgment remains unclear, and we cannot trace back the decision-making process based on the retrieval results.

Therefore, we aim to explicitly integrate judgment prediction with legal document retrieval. However, there remain the following challenges to achieve our goal. Firstly, legal document retrieval and judgment prediction are usually formulated as two distinct machine learning problems–retrieval and classification. It is difficult for one retrieval model to predict the applicable judgment for legal cases and in turn leverage the judgment prediction to enhance retrieval. Secondly, legal documents are inherently lengthy and complicated. It results in the retrieval efficiency issue and hiders to represent each document as a shared and informative representation for both tasks. Thirdly, both tasks rely on specialized law knowledge [18, 31], an appropriate way that effectively injects law expertise to guide the prediction of both tasks remains a concern.

Facing the above challenges, we propose a novel law-guided generative legal document retrieval method, namely GEAR. GEAR explicitly integrates judgment prediction with legal document retrieval in a sequence-to-sequence (Seq2Seq) manner as illustrated in Figure 1. The insight of GEAR lies in formulating the retrieval process into the generation of law-aware semantic IDs, where each ID not only represents a document relevant to the query but also reflects its applicable judgments. Specifically, we first construct a corpus based on the definition of charges in law[1] and subsequently extract rationales [32, 42] representing the key elements/ circumstances [21, 30] at the word- and sentence-level for each document according to this corpus. We employ these rationales instead of raw documents as queries for generation. This strategy not only serves to effectively filter out the noise of legal documents, ensuring generation efficiency but also renders the rationales shared and informative for both tasks. Then, we create the law structure constraint tree based on the inherent hierarchy of law (e.g. Chapter-Section-Article), considering that both tasks are learned with the guidance of law. Given this tree, we assign legal documents hierarchical semantic IDs, with the IDs reflecting their judgments, for example, the ID "0-2-5-269-809" indicates the document named 809 falls under Article 269 of Chapter 5 of Section 2. In this way, the generation of these IDs is equivalent to traversing the tree from the root through intermediate judgment nodes, to document-specific leaf nodes. It makes GEAR capable of showing the legal reasoning process and performing dual predictions for judgment and relevant documents in a single inference. To further improve the accuracy and consistency of both tasks, we devise a novel training objective called the revision loss. This loss aligns with the hierarchy of the tree and jointly minimizes the discrepancy between predicted and labeled judgments/ retrieved cases. Extensive experiments on two Chinese legal case retrieval datasets show the consistent superiority of GEAR over state-of-the-art methods while maintaining competitive judgment prediction performance. We also validate the effectiveness of GEAR on a French statutory article retrieval dataset, reaffirming its generalization ability.

The major contributions of the paper are summarized as follows:

(1) To the best of our knowledge, this is the first work that **explicitly** integrates judgment prediction with legal document retrieval. Our method is capable of showing the legal reasoning process and performing dual predictions for both tasks in a single inference. It improves the transparency of the legal decision-making.

(2) We propose a novel law-guided generative model, namely GEAR. We explicitly leverage the law knowledge to extract rationales from legal documents, assign them the law-aware hierarchical IDs, and formulate the prediction as a traversal on the law structure constraint tree. We also propose the revision loss to jointly improve the accuracy and consistency of both tasks.

(3) We conduct extensive experiments on three public datasets of legal document retrieval in two languages. The results indicate GEAR not only achieves state-of-the-art performance in legal case and statutory article retrieval but also maintains competitive judgment prediction performance.

2 RELATED WORK

2.1 Legal Document Retrieval

Legal document retrieval is a long-standing research topic in the field of information retrieval. In the early exploration, researchers [4, 14, 23, 27, 43] made efforts to inject the legal knowledge to the retrieval through the decomposition of legal issues and involving the ontology. In recent years, deep learning demonstrated its effectiveness in exploring document semantics. One representative line of works focused on the network-based precedents methods tailored for the common law systems. For example, Minocha et al. [22] leveraged the Precedent Citation Network (PCNet) to predict the

[1] In this paper, we take laws in China and Belgium as examples. It is worth noting that GEAR can readily accommodate laws from various other countries.

relevance based on whether the sets of precedent citations occur in the same cluster. Bhattacharya et al. [5] proposed Hier-SPCNet to capture all domain information inherent in laws and precedents. The other line of works judged the relevance between the query and candidate document according to their text-level similarity. One such method was BERT-PLI [29]. It divided the legal document into several paragraphs and used BERT [10] to obtain the similarity between the paragraphs. Lawformer [38] was another text-based method. It used millions of Chinese criminal and civil case documents to pre-train a Longformer [3] model. Despite impressive performance, these works overlook the significance of judgment prediction and merely focus on the text-level similarity, resulting in sub-optimal and unreliable results.

Recently, Li et al. [15] introduced an implicit training objective that uses the fact description of the legal document to predict its judgment, expecting a proper alignment of legal documents in vector space based on their judgments. It fails to provide explicit evidence of judgment consistency for relevance modeling, leading to inaccuracies and a lack of transparency.

2.2 Generative Retrieval

Generative retrieval has recently emerged as a promising direction for document retrieval. These methods assign semantic IDs to documents and utilize language models for ID generation. It enables an end-to-end retrieval in contrast to the traditional index-then-retrieve paradigm. Various methods have been introduced to generate semantic document IDs. For example, Cao et al. [6] introduced a Seq2Seq system to conduct entity retrieval. They first represented documents as unique names that are composed of entity names. Then they used a auto-regressive generation model to generate the unique names of these entities based on contextual information. Tay et al. [34] proposed the differentiable search index (DSI) paradigm, which is an auto-regressive generation model to perform ad-hoc retrieval tasks. The input of the model was a natural language query and the model regressively generated documents' ID strings that are relevant to the given query. Wang et al. [35] proposed a novel method NCI, which used a tailored prefix-aware weight-adaptive decoder to optimize the retrieval performance. Ultron [48] leveraged document titles and substrings as IDs to enrich the semantic information of IDs. To mitigate data distribution mismatch that occurs between the indexing and the retrieval phases, Zhuang et al. [49] proposed DSI-QG, which adopted a query generation model with a cross-encoder to generate and select a set of relevant queries.

Existing studies focused on the general domain, lacking specific designs for legal documents and the integration of law knowledge.

3 METHODOLOGY

3.1 Task Formulation

In this work, we target on legal document retrieval including legal case retrieval (LCR) and statutory article retrieval (SAR). Suppose that we have a set of collected samples $\mathcal{D} = \{(q, C, \mathcal{R})\}$. For each data instance, q is the query representing an undecided legal case submitted by the legal practitioner in LCR, a legal question in SAR; $C = \{c_1, c_2, \cdots, c_N\}$ with size $N \in \mathbb{N}^+$ is the candidate precedent case set in LCR, the statutory article pool in SAR; $\mathcal{R}$ represents the labeled relevant case/ statutory article set from C given the query q. Unlike previous studies that only predict to retrieve $\mathcal{R}$ from C given q, in this work, we instead unify judgment prediction and document retrieval into a generative retrieval framework, and thus aim at learning a retrieval function $f : q \times C \rightarrow \mathcal{R} \times \mathcal{E}$, where $\mathcal{E}$ denotes the set of applicable judgment corresponding to q.

3.2 Overall Framework

To learn f and explicitly integrate judgment prediction with legal document retrieval, we develop GEAR, a novel law-guided generative approach from the viewpoint of generative retrieval. Essentially, given a query document, GEAR adopts a language model to perform the Seq2Seq generation where the retrieved documents are represented as semantic IDs. Following the practice of [34, 35, 48, 49], GEAR consists of two major steps to directly generate IDs of documents as the retrieval target. In the first indexing step that focuses on memorizing the information about each document, our GEAR takes each document c as input and generates its ID id^c as output. The model is trained with the standard language model objective with the teacher forcing:

$$\mathcal{L}_i = \sum_{c \in C} \log P(id^c | c). \tag{1}$$

In the second retrieval phase, GEAR models associate each q to its relevant document r through an auto-regressive generation:

$$\mathcal{L}_r = \sum_{(q,r) \in \mathcal{D}} \log P(id^r | q), \tag{2}$$

where id^r denotes the ID of r. As such, once a GEAR model is trained, it can be used to retrieve candidate documents for a test query in an end-to-end manner using beam search.

As aforementioned in Section 1, there are several challenges in the legal domain. For explicitly integrating judgment prediction with legal document retrieval within GEAR, the critical learning tasks become: (1) extract rationales instead of using raw documents to form the input for generation (**Section 3.3**); (2) create informative law-aware IDs for each document based on the hierarchical structure of law (**Section 3.4**); (3) develop a training objective to explicitly ensure judgment-level and document-level consistency between predictions and labels (**Section 3.5**).

3.3 Rationale Extraction

To ensure efficiency and provide shared and informative representations for legal documents, based on the set of laws L, we devise a module called f_R to extract rationales E instead of using the raw document $doc \in \{q, c\}$ as the input of GEAR:

$$E = f_R(doc, L), \tag{3}$$

where $E = \{E_w, E_s\}$ in which E_w and E_s respectively denotes the word-level and sentence-level rationales.

As illustrated in Figure 2, we first leverage the guidance of law to construct a corpus $B = \{B_1, B_2, B_3\}$, collecting law-based keywords B_1, B_2 and embeddings B_3. The keyword set B_1 is constructed according to the lexical variants of all charge names in L. For example, in terms of the charge "crime of forges the seals of a company, enterprise, institution or a people's organization"(translated from Chinese), we split the charge name and remove the stop words,

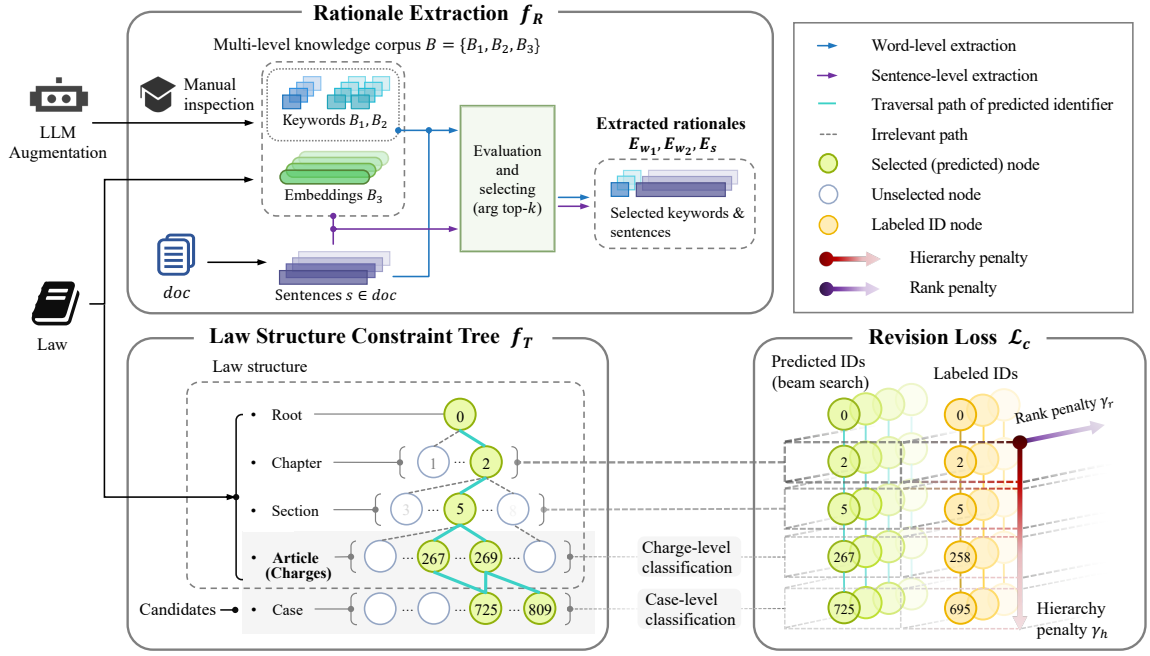


Figure 2: The proposed three modules of GEAR. f_R extract rationales from legal documents; f_T assigns hierarchical IDs to each document and constrain the decoding; $\mathcal{L}_c$ jointly optimizes document retrieval and judgement prediction.

then add the rest and their lexical variants to B_1. The keyword set B_2 is constructed similarly to the B_1. We split the definitions of charges in law and remove the stop words then add the rest and their lexical variants to B_2. For the augmentation purpose, we employ the definitions of charges in law as prompts for the large language model[2] (LLM) designed for the legal domain. After obtaining feedback from LLM, we remove stop words and incorporate the results into B_2. To avoid the hallucination issue, we engage legal experts to manually assess the quality of augmentations to ensure the effectiveness of B_2. B_3 is constructed by collecting LegalBERT [7, 10] embeddings for the definition of each charge in law.

Once the corpus is collected, we compute the multiple-level scores and extract E for each document based on three corpora as follows. At the word level, we split the document and respectively select the top-k_1 (top-k_2) keywords E_{w_1} (E_{w_2}) from B_1 (B_2) as follows:

$$E_{w_i} = \underset{w \in B_i}{\arg\,\text{top-}k_i} \Big(\text{tf}(w, doc) \Big), \tag{4}$$

where $i \in \{1, 2\}$; w is the word from B_i; k_1 and k_2 are hyperparameters to control the number of selected words; tf$(\cdot)$ denotes term frequency. At the sentence level, for each sentence s in doc, we first select E_s from as follows:

$$E_s = \underset{s \in doc, l \in L}{\arg\,\text{top-}k_3} \Big(\lambda_1 \frac{\text{sim}(s, B_1)}{\text{len}(s)} + \lambda_2 \frac{\text{sim}(s, B_2)}{\text{len}(s)} + \lambda_3 \cos(\text{emb}(s), \text{emb}(l)) \Big), \tag{5}$$

where k_3 is another hyperparameter to control the number of selected sentences; $\lambda_1, \lambda_2, \lambda_3$ are balance coefficients; len() denotes the sentence length; $\cos(\cdot)$ denotes the cosine similarity; emb() denotes the embedding function; sim$(\cdot)$ is defined as:

$$\text{sim}(s, B_i) = \sum_{w \in s} \text{tf}(w, B_i), \tag{6}$$

where w is the word from s. Please note that since query documents are typically undecided i.e., without labeled applicable charges, we extract rationales using the method described above. For the candidate precedent documents, whose applicable charges are given, we shrunk B into a corpus constructed based on their corresponding labeled charges.

3.4 Law Structure Constraint Tree

Given that both legal document retrieval and judgment prediction require guidance by law, typically organized in a "Chapter-Section-Article" hierarchy, we argue that the decision process to judge whether two documents are relevant in document retrieval is analogous to the search in such a tree-like hierarchy. In other words, when legal practitioners search for the relevant documents given a query, they always expect the charge of relevant documents to be located at the same position in the law hierarchy as the charges applicable in the query. Therefore, we devise a module f_T that leverages the inherent hierarchy of the law to construct a law structure constraint tree T as illustrated in Figure 2 and assigns the law-aware semantic ID id for each $c \in C$:

$$id = f_T(c, T), \tag{7}$$

[2]We use ChatLaw [9] for Chinese data. Since we have not found a suitable LLM for the Belgian legal domain, we omit this step for this data.

where *id* is in the prefix-suffix style. The prefix depends on the position of the applicable charges within the tree. As shown in Figure 2, document 809 involves the charge of "crime of robbery" which falls under Article 269 of Chapter 5 of Section 2 of the Criminal Law of the People's Republic of China. Hence, the assigned prefix for this document is 0-2-5-269, where 0 represents the root node of the tree. As for the suffix, we regard documents are the children of their corresponding charges on the tree, and assign a unique ID to each of them under a crime node. Compared to current works that employ hierarchical k-means to create IDs for each document, ours avoid same integers may have different meanings at different levels, and thus ensure the effectiveness of model training.

On the other hand, one unique feature of the legal domain is that a single document can involve multiple charges. When the query is this kind of document, the ideal retrieval results should encompass these charges. Therefore, we assign k IDs to documents involving k charges, with each ID corresponding to a specific charge. For example, as shown in Figure 2, the 725 involves charge 267 and charge 269, the valid IDs for this document include 0-2-5-267-725 and 0-2-5-269-725. In doing so, during retrieval, we expect the model to retrieve all IDs of the target documents, thereby increasing the probability of the target being retrieved.

3.5 Revision Loss

Besides employing the typical language model training objectives (Equation 1 and Equation 2), we also develop a novel training objective called the revision loss for the consistency between the query and retrieved documents, i.e., we aim to directly minimize their judgment-level and document-level discrepancy.

Formally, as illustrated in Figure 2, given the predicted ID (list of integers) $[\hat{id}_1, \hat{id}_2, \cdots, \hat{id}_L]$ and the corresponding ground-truth ID $[id_1, id_2, \cdots, id_L]$ for a query q, both having a length of L, we compare the difference between them and calculate the reward R_t at each step $t \in [1, L]$ as follows:

$$R_t = \begin{cases} \mu, & \text{if } \hat{id}_t = id_t, \\ -\gamma_h^{L-t}\mu|\hat{id}_t - id_t|, & \text{if } \hat{id}_t \neq id_t, \end{cases} \tag{8}$$

where μ is the constant reward unit, $\gamma_h \in (0, 1]$ is the hierarchy penalty factor used to penalize the differences between predictions and labels layer by layer along the law structure constraint tree, with larger penalties as it gets closer to the tree root and smaller penalties as it gets closer to the tree leaves. Intuitively, if two documents share the same prefix, they are likely relevant to each other because of the same applicable charge, receiving higher rewards.

Then we apply the REINFORCE algorithm [36] to optimize the model parameters, the revision loss is defined to minimize the policy gradient objective:

$$\mathcal{L}_c = -\sum_{q\in\mathcal{D}}\sum_{b}^{bz}\gamma_r^b\sum_{t}^{L}\log p(\hat{id}_t|q)\cdot R_t - \lambda_w \log p(id_t|q), \tag{9}$$

where $\gamma_r \in (0, 1]$ the optional penalty factor used to focus on the top of retrieved document list; bz denotes the beam size, i.e., the number of documents to retrieve for each query; $p(\hat{id}_t|q)$ is the probability that predicting to generate $\hat{id}_t$ given q at layer t; λ_w is the balance coefficient. To further handle the sparse reward issue and improve the training efficiency, we follow [8, 17, 39] and add the second term to Equation 9 that directly increases the probability of generating id_t. Thus, the overall training objective is:

$$\mathcal{L} = \mathcal{L}_i + \mathcal{L}_r + \lambda_l\mathcal{L}_c, \tag{10}$$

where λ_l is the coefficient to balance the indexing loss (Equation 1), retrieval loss (Equation 2), and the revision loss.

3.6 Inference

In the inference, we aim to retrieve the top-k documents from the candidate pool. Since we have assigned hierarchical semantic ID to each document based on the law-aware constraint tree where each leaf node corresponds to a candidate document in the pool, we utilize the constrained beam search [1, 11, 25] to ensure all the generated document IDs are valid within the tree.

4 EXPERIMENTS

In this section, we conduct experiments to answer the following research questions: **RQ1:** How does GEAR perform on legal document retrieval compared to state-of-the-art methods? **RQ2:** How effective are the three modules in GEAR? **RQ3:** Can GEAR show competitive performance on judgment (applicable charges) prediction? **RQ4:** What is the quality of the rationales extracted by GEAR including effectiveness and efficiency? **RQ5:** Can GEAR incur less time overhead in legal document retrieval compared to popular generative methods? **RQ6:** How robust is GEAR across languages and domains (e.g. in statutory article retrieval)?

The source code and datasets have been shared at: https://github.com/E-qin/GEAR.

4.1 Experimental Settings

4.1.1 Datasets. **ELAM**[3] [42] is a Chinese LCR dataset, focusing on criminal cases. ELAM has corresponding labels for both case retrieval and judgment prediction, which is suitable for our goal. We exclude those cases with multiple applicable charges to consider the retrieval and judgment prediction performance in a single charge scenario. The resulting candidate pool size of ELAM is 1332. Other data preprocessing is aligned to [15].

LeCaRDv2[4] [16] is the official updated version of LeCaRD [21]. In this dataset, the relevance labels are divided into four levels, ranging from 3 to 0, indicating a gradual decrease in relevance. We follow the data preprocessing approach of [15], with the exception of increasing the candidate pool size from 100 to 1390 to further validate the effectiveness of baselines and our model. In LeCaRDv2, cases encompass both single and multiple charges, averaging 1.5 charges per case. Considering the ground-truth judgment labels of query cases have not been provided, we ask two legal experts (Ph.D. in Law) to annotate the charge label for the testing queries. The experts are proficient in Chinese criminal law with sufficient experience in handling cases similar to this dataset. They carefully align the LeCaRDv2 judgment criteria [16] before annotation and discuss opinions to reach a consensus, ensuring accurate labeling.

BSARD[5] [19] is a SAR dataset composed of more than 1.1K legal questions labeled by domain experts with relevant articles selected

[3] https://github.com/ruc-wjyu/OPT-Match.
[4] https://github.com/THUIR/LeCaRDv2.
[5] https://huggingface.co/datasets/maastrichtlawtech/bsard.

from the 22K law articles gathered from 32 publicly available Belgian codes. It is worth noting that BSARD contains structural annotations of corresponding laws, facilitating the utilization of law structural knowledge.

4.1.2 Baselines. We consider three types of baselines in this study.

(1) Sparse retrieval methods: Query Likelihood (QL) [44] is a probabilistic language modeling approach employed to assess the relevance of documents to a provided query. **BM25** [26] is a probabilistic information retrieval model widely used in the field of text retrieval. BM25 takes into account both term frequency and document length normalization.

(2) Dense retrieval methods: BERT [10] is a strong baseline in ad-hoc retrieval tasks in the open domain. In this paper, we adopt the checkpoint that is pre-trained on a large Chinese corpus Following [34, 35, 49], after encoding legal documents using BERT, we then apply Approximate Nearest Neighbor (ANN) search algorithms to retrieve relevant documents. **Legal-BERT**[6] [7] is a variant of BERT that undergoes specific training in the legal domain to better understand and process text related to law. **Lawformer** [37] is a Longformer [3] backbone pre-trained on large legal case corpus, to encode legal texts. **ChatLaw-Text2Vec**[7] [9] is a legal text matching model based on ChatLaw which is pre-trained on a corpus of 936,727 legal documents. **G-DSR** [20] uses legal-CamemBERT[8], a legal variant of CamemBERT trained on BSARD dataset. It takes into account both the dense representation of text and the graph representation of legal structures. G-DSR is the state-of-the-art SAR method in the French legal domain. **SAILER** [15] is a structure-aware LCR model. It adopts an asymmetric encoder-decoder architecture to integrate structures of legal case document information into dense vectors. SAILER achieves state-of-the-art retrieval performance in Chinese LCR domain. As for the training of all baselines, we follow [34, 35, 49] and use Approximate nearest neighbor Negative Contrastive Estimation (ANCE) [40] method.

(3) Generative retrieval methods: DSI [34] is a new paradigm for document retrieval tasks. It utilizes a Transformer-based encoder-decoder model to map queries directly to relevant IDs. DSI achieves the end-to-end retrieval. **NCI** [35] improves DSI in terms of using constrained beam search and prefix-aware weight-adaptive decoder. Both DSI and NCI use hierarchical k-means clustering of document vectors to create k-means IDs. **DSI-QG** [49] design a query generation process on the top of DSI, which can mitigate data distribution mismatches present between the indexing and the retrieval phases. **Ultron** [48] improves DSI through adopting product quantization to create semantic IDs and using URLs to create term-based IDs.

4.1.3 Implementation Details. We implement baseline methods following the suggestions in the original papers.

(1) For classical term-based baselines, we use the pyserini and genism toolkits with the default parameters.

(2) For dense retrieval baselines, we use the open-sourced checkpoint to initialize model parameters of the pre-trained models and use faiss toolkit to implement ANN algorithms. The batch size is set to 16. The max length of the input text is set to 1024 for Lawformer and 512 for the other models. We tune these models with in-batch contrastive loss.

(3) For generative retrieval baselines, we directly use their official open-source implementations, employing the pre-trained T5 "Randeng"[9] as backbone for Chinese legal domain, and "t5-base"[10] for the French legal domain. We use beam search to retrieve relevant cases, where the beam size is set to 30.

We keep the backbone model and beam size same as baselines and set the max input length of GEAR to 512, the rest hyperparameters are tuned as follows: the batch size is set to 2; the learning rate is tuned from [1e-5,1e-4] with step size 2e-5; in the rationale extraction module, k_1, k_2, k_3 are respectively set to 2, 5, 15 for ELAM and 10, 20, 15 for LeCaRDv2; λ_1, λ_2, λ_3 are respectively set to 10.0, 1.0, 0.1; in the law structure constraint tree module, the height of the tree (the length of the hierarchical ID) L is set to 4; in the training, μ is set to 1; γ_h is tuned from $\{0.01, 0.1\}$; γ_h is set to 1; λ_w is tuned from $\{1, 10, 100\}$; λ_l is tuned from [1e-4,1e-2] with step size 5e-4. Hyperparameters of GEAR are tuned using grid search with Adam[13]. ALL experiments are conducted on a single NVIDIA RTX A6000.

Table 1: Statistics of the datasets. Since queries in BSARD do not involve judgments, we omit the judgment prediction comparison on this dataset.

Statistics	Dataset ELAM	LeCaRDv2	BSARD
Avg. length per candidate document	1163.68	1568.38	880.29
Avg. length per query document	1304.98	558.18	92.48
Avg. # charges per candidate case	1.00	1.50	-
# Available candidates per query	1332	1390	1612
# Query documents involved	147	653	1108
# Charges involved of judgment	97	100	-

4.1.4 Evaluation Metrics. For a fair comparison, we follow previous works [34, 35, 48, 49] and leverage the commonly adopted metrics, including Recall (R) and MRR. The averaged results on all test cases are reported.

To demonstrate GEAR's ability on judgment prediction, we introduce a new metric called *coverage@k*, assessing the percentage of charges involved in the query that are covered by top-k retrieved documents. This metric evaluates the charge-level consistency, i.e., the extent to which the retrieval process contributes to the efficacy of legal judgment. Formally, *coverage@k* is defined as:

$$coverage@k = \frac{1}{N_q} \sum_{i=1}^{N_q} \frac{|\mathcal{E}_i^{\text{Top-}k} \cap \mathcal{E}_i|}{|\mathcal{E}_i|}, \tag{11}$$

where $|\ |$ is the size of a set; for i-th testing query, $\mathcal{E}_i$ denotes its label set of applicable judgment charges, $\mathcal{E}_i^{\text{Top-}k}$ denotes the the set of charges of the top-k retrieved documents; N_q is the number of testing queries. In our experiments, we compute *coverage@k* metric at 1, 3, 5, and 10.

[6] https://github.com/thunlp/OpenCLaP.
[7] https://huggingface.co/chestnutlzj/ChatLaw-Text2Vec.
[8] https://huggingface.co/maastrichtlawtech/legal-camembert-base.
[9] https://huggingface.co/IDEA-CCNL/Randeng-T5-77M-MultiTask-Chinese.
[10] https://huggingface.co/t5-base. Please note that this is applicable to French datasets.

Table 2: Performance comparisons of our approach and the baselines on ELAM dataset and LeCaRDv2 dataset. The best and the second-best performances are denoted in bold and underlined fonts, respectively. "R@K" is short for "Recall@K". † denotes GEAR performs signifcantly better than baselines based on two-tailed paired t-test with Bonferroni correction ($p < 0.05$).

	ELAM					LeCaRDv2				
Models	**R@1**	**R@5**	**R@10**	**R@20**	**MRR**	**R@1**	**R@5**	**R@10**	**R@20**	**MRR**
QL [44]	0.0272	0.1088	0.1361	0.2857	0.0723	0.0252	0.0892	0.1351	0.2177	0.1327
BM25 [26]	0.0340	0.0680	0.1497	0.2245	0.0635	0.0435	0.1452	0.2549	0.3900	0.1862
BERT [10]	0.0302	0.1008	0.1405	0.2861	0.1521	0.0299	0.0718	0.1557	0.2534	0.1162
Legal-BERT [47]	0.0384	0.0938	0.1509	0.3123	0.1510	0.0218	0.0620	0.1081	0.2743	0.1138
Lawformer [37]	0.0537	0.1682	0.2220	0.3789	0.1701	0.0518	0.1491	0.2728	0.3593	0.1638
ChatLaw-Text2Vec [9]	0.0385	0.1371	0.2065	0.3323	0.1694	0.0356	0.0813	0.1510	0.3380	0.1379
SAILER [15]	0.0729	0.2132	0.3282	0.4604	0.2029	0.0608	0.1644	0.2910	0.4271	0.2018
DSI [34]	0.0204	0.1134	0.2274	0.3159	0.1249	0.0232	0.0577	0.0768	0.1285	0.1159
DSI-QG [49]	0.0278	0.1606	0.2736	0.3803	0.1531	0.0283	0.0725	0.1230	0.1881	0.1224
NCI [35]	0.0325	0.0936	0.1463	0.2070	0.1285	0.0416	0.1024	0.1696	0.2504	0.1914
Ultron [48]	0.0607	0.1583	0.2260	0.3506	0.1678	0.0333	0.1207	0.2142	0.3492	0.1511
GEAR	**0.0793**†	**0.2368**†	**0.3356**†	**0.4976**†	**0.2365**†	**0.0630**†	**0.1706**†	**0.3142**†	**0.4625**†	**0.2162**†
w/o $\mathcal{L}_c$	0.0611	0.1657	0.2793	0.4167	0.1763	0.0452	0.1485	0.2549	0.4478	0.1979
w/o f_R	0.0586	0.1830	0.2913	0.4437	0.1802	0.0308	0.1143	0.2108	0.3714	0.1621
w/o f_T	0.0464	0.1429	0.2328	0.3158	0.1626	0.0166	0.0550	0.0971	0.1388	0.1596

4.2 Retrieval Performance

Table 2 presents the retrieval performance of GEAR and baselines on the ELAM and LeCaRDv2. All the methods are trained 10 times and the averaged results are reported. From the results, we have the following observations for **RQ1**:

(1) **GEAR demonstrates a significant performance advantage over all baseline methods on both datasets.** The relative improvements of MRR on the ELAM and LeCaRDv2 are at least 16.55% and 7.13%, respectively. These results indicate GEAR's effectiveness. We attribute the improvement to the elaborate design for explicitly integrating judgment into case retrieval including rationale extraction, law-aware ID assignment, and the revision loss. See Section 4.3 for the detailed analysis of each module of GEAR. (2) **Compared to sparse retrieval and dense retrieval methods, current generative retrieval methods struggle to achieve satisfying performance in the legal domain.** Without injecting the law knowledge, generative retrieval baselines inevitably yield sub-optimal results. On the other hand, SAILER introduces an implicit training objective that uses the fact description of the legal document to predict its judgments. In this way, SAILER produces judgment-aware document representations and achieves the second-best performance. However, SAILER fails to provide explicit evidence of judgment consistency for relevance modeling, leading to an inferior retrieval performance compared to GEAR.

4.3 Ablation Studies

To answer **RQ2**, we conduct an ablation study to investigate the impact of each component of GEAR.

Firstly, we test the performance of GEAR's variants by removing a certain component. Following DSI [34], we replace rationale extraction and law-aware hierarchical IDs with Direct indexing (using the first 512 tokens of document as queries for language

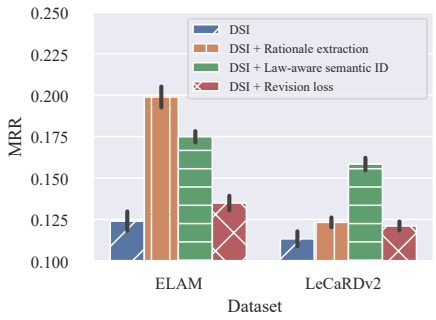
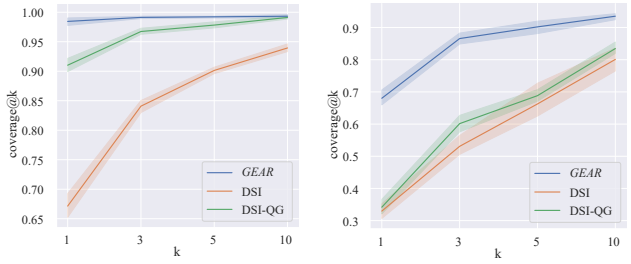


Figure 3: Comparison of retrieval performance of DSI and DSI equipped with the proposed three modules. The means values of 5 repeated experiments are reported, with error bars representing the 95% confidence interval of the means.



(a) *coverage* on ELAM (b) *coverage* on LeCaRDv2

Figure 4: Comparison of the judgment prediction performances in terms of *coverage*. The proposed GEAR consistently outperforms DSI and DSI-QG in both single (ELAM) and multiple (LeCaRDv2) charges scenarios.

models) and k-means IDs. As shown in the bottom part of Table 2, we observe that: (1) Removing modules $\mathcal{L}_c$, f_R, and f_T individually results in a performance degradation of 25.51%, 23.83%, and 31.16% on ELAM, 8.50%, 25.00%, and 26.13% on LeCaRDv2. These results verify the effectiveness of all three modules. (2) It is worth noting that removing our law-aware hierarchical IDs results in the most significant performance decrease. It is because we inject law knowledge into each digit of the ID, aligning them with the hierarchical structure of laws. By simulating the retrieval to mirror the legal decision-making process, we enable GEAR to effectively learn the association between query cases and candidate cases. These results demonstrate the effectiveness of our IDs and highlight the importance of introducing structural and semantic knowledge in law to legal document retrieval.

To further validate the efficacy of the components of GEAR, we combine the three components of GEAR with DSI individually and test its retrieval performance. We conduct 5 repeated experiments and present the mean values of MRR, with error bars representing the 95% confidence interval. As illustrated in Figure 3, integrating the components of GEAR into DSI consistently results in performance improvements. Specifically, DSI equipped with rationale extraction achieves an improvement of over 60% on ELAM, and DSI equipped with law-aware IDs demonstrated a 42% improvement on LeCaRDv2 dataset. However, DSI equipped with revision loss does not exhibit a substantial improvement. Because DSI utilizes k-means IDs and the length of these IDs may vary due to differences in cluster sizes. In such cases, our revision loss cannot accurately measure the differences between predictions and labels, resulting in a slight performance improvement.

4.4 Quality of Judgment Prediction

To answer **RQ3**, we conduct experiments to evaluate the judgment prediction accuracy of retrieved cases using the proposed metric *coverage@k*. Please note that we consider the applicable charges as the judgment results.

We run experiments ten times and present the results in Figure 4, reporting the average score with the shaded area indicating the 95% confidence interval. From the plot, we observe that: (1) in the single-charge scenario (ELAM) as shown in Figure 4(a) , GEAR achieves a remarkably high *coverage* score (over 0.95) with just 1 case retrieved. In the multi-charge scenario (LeCaRDv2) as shown in Figure 4(b), GEAR retrieves about 3 cases to encompass approximately 85% of the charges. These results demonstrate that GEAR has considerable ability in charge prediction. This capability stems from our integration of judgment prediction into case retrieval, unifying the predictions for both tasks in a traversal on the law structure constraint tree. (2) GEAR demonstrates a significantly *coverage* improvement compared to DSI and DSI-QG on both datasets, especially when a limited number of cases are retrieved, such as 1 or 3. This mainly attributes to the specialized design of GEAR for judgment prediction tasks including the law-aware hierarchical IDs and the revision loss, which explicitly enhances the accuracy of judgment predictions. These results verify that GEAR is capable of performing competitive legal judgment predictions.

Table 3: Time overhead and human evaluation of rationale extraction over 50 random samples from ELAM and LeCaRDv2 by two annotators with the inter-rater agreement of 0.96. "Time"(ms) denotes the extraction time per sample.

Methods	ELAM		LeCaRDv2	
	Acc.	Time	Acc.	Time
Direct indexing [34]	0.66	-	0.38	-
Doc2query [49]	0.48	4094.78(±1363.07)	0.26	3947.01(±1229.81)
ChatLaw [9]	0.94	20975.92(+3858.29)	0.88	21827.05(±5820.58)
f_R in GEAR	0.94	**0.10(±0.01)**	0.86	**0.67(±0.03)**

4.5 Effectiveness and Efficiency of Rationale Extraction

To answer **RQ4**, we conduct experiments and evaluate the rationales extracted by GEAR in comparison to those generated by prevalent query generation methods featured in existing generative retrieval studies. We consider the following baselines: Direct indexing [34], which means using the first 512 tokens of the raw document as the query; Doc2query [35, 49], which uses a language model to generate pseudo text as the model input; and ChatLaw [9], which demonstrates to provide legal summaries that are on par with human-level quality.

First, we follow the practice [42, 46] and conduct the human evaluation to assess the quality of extracted rationales. We randomly selected 50 samples from both ELAM and LeCaRDv2 and asked two annotators (Ph.D. in Law) to determine whether the extracted rationales (generated queries) are sufficient to ascertain the applicable charges for the original cases, indicating that the extracted sentences comprehensively encompass the primary information from the original cases. Each annotator is provided with a pair of the rationales (generated queries) and the corresponding original case for each sample and is asked to mark the sample as 1 if they agreed with it and 0 otherwise. We calculate the accuracy of the sample rationales based on annotators' evaluation. As shown in Table 3, the results illustrate rationales extracted by GEAR consistently outperform both Direct indexing and Doc2query on two datasets, achieving performance comparable to that of ChatLaw. These results confirm the rationales extracted by GEAR are informative for both tasks.

Then, we conduct an experiment to validate the time overhead of rationale extraction. Since Direct indexing takes the first 512 tokens as the query, we omit its time overhead. From the results shown in Table 4, we can see that GEAR achieves impressive accuracy on par with ChatLaw but with far less time consumption. Compared to Doc2query, GEAR also exhibits a significant advantage in terms of time overhead. These results verify the GEAR's efficiency in terms of rationale extraction. Based on the results, the generative approaches for rationales are not advisable for legal document retrieval due to the efficiency issue.

4.6 Inference Time and Convergence

To answer **RQ5**, we record the inference wall time (in milliseconds) per query of baseline methods and GEAR. This comparison is conducted on a single NVIDIA RTX A6000 across two datasets. From the results shown in Table 4, we observe that GEAR demonstrates

Table 4: Comparison in average inference wall time (95% confidence interval) per query on a single NVIDIA RTX A6000.

Models	Dataset	
	ELAM	LeCaRDv2
Ultron	110.544(±24.703)	31.854(±7.766)
DSI-QG	73.147(±17.062)	26.369(±5.973)
DSI	72.646(±14.820)	27.023(±6.084)
GEAR	**59.184(±8.737)**	**20.391(±4.064)**

Table 5: Performance comparisons of our approach and the baselines on BSARD. "R@K" is short for "Recall@K". † indicates that improvements are significant based on two-tailed paired t-test with Bonferroni correction ($p < 0.05$).

Models	R@5	R@10	R@20	R@30	MRR
QL [44]	0.1752	0.2128	0.2698	0.2923	0.1770
BM25 [26]	0.1815	0.2381	0.2871	0.3056	0.1786
G-DSR [20]	0.1636	0.3426	0.5081	0.6720	0.3012
DSI [34]	0.4578	0.5536	0.6315	0.6562	0.3955
GEAR	**0.5297**†	**0.6164**†	**0.7031**†	**0.7170**†	**0.4081**†

superior efficiency, with an inference time of 59.184 ms (±8.737) for the ELAM dataset, and 20.391 ms (±4.064) for LeCaRDv2. Specifically, when compared to Ultron, GEAR achieves a remarkable 46.48% reduction in inference time for the ELAM dataset and a 36.03% reduction for LeCaRDv2. This is mainly because Ultron uses product quantization to create the IDs for documents. For all document embeddings, Ultron first divides embedding space into several groups and then performs k-means clustering on each group. It usually leads to excessively long IDs. In the case of DSI and DSI-QG, GEAR exhibits a substantial 19.01% and 18.59% improvement for ELAM and a 22.64% and 18.59% improvement for LeCaRDv2. In DSI and DSI-QG, the tree constructed by k-means may be unbalanced, meaning that the lengths of case IDs are unequal. Some case IDs may be longer, which impairs the inference performance.

For further confirming GEAR's efficiency, we plot the testing curves for DSI, DSI-QG, and our GEAR with the x-axis denoting the number of epochs, the y-axis denoting the MRR score, and the shaded area indicating the 95% confidence interval. As illustrated in Figure 5, we observe that GEAR not only outperforms DSI and DSI-QG significantly in terms of performance but also exhibits superior convergence. GEAR achieves near-optimal MRR performance by 6 epochs, whereas DSI-QG and DSI converge at 8 and 10 epochs, respectively. The results verify the efficiency of GEAR.

4.7 Robustness across Languages and Domains

To answer **RQ6**, we compared the performance of baselines and GEAR on BSARD, a French SAR dataset. Since BSARD has a relatively short query length, we omit the rationale extraction part and take the raw legal questions as queries for GEAR. In terms of the ID, we follow the structure of the Belgian code provided in the data and assign the hierarchical semantic to each statutory article. From the results shown in Table 5, we have two observations: (1) with a small number of documents to retrieve, generative retrieval

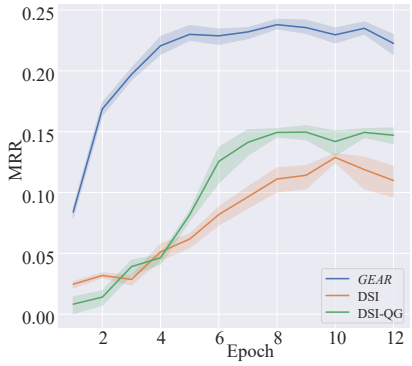


Figure 5: Testing curves of DSI, DSI-QG, and our GEAR.

methods (DSI and GEAR) exhibit significantly higher retrieval performances in SAR scenario compared to dense retrieval. We assume the reason why dense models perform poorer is that there exists a significant gap between legal questions and statutory articles. It is difficult for dense models to learn the correct association between them especially without the law knowledge injected. (2) GEAR demonstrates the best performance, exhibiting a substantial advantage over sparse and dense retrieval methods including the current state-of-the-art model G-DSR. The improvement of GEAR benefits from explicitly injecting legal knowledge into generative retrieval frameworks.

5 CONCLUSION

In this study, we introduce GEAR, a novel law-guided generative legal document retrieval method that explicitly integrates judgment prediction. GEAR exploits the law knowledge and extracts rationales from legal documents, ensuring a shared and informative representation for both tasks. Grounded in the inherent hierarchy of laws, GEAR constructs a law structure constraint tree and assigns the law-aware semantic ID to each document. These designs enable a unified traversal from the root, through intermediate charge nodes, to case-specific leaf nodes, which empowers GEAR to perform dual predictions for judgment and relevant documents in a single inference. With the help of the proposed revision loss, GEAR jointly minimizes the discrepancy between the IDs of predicted and labeled judgments/ retrieved documents, improving the accuracy and consistency for both tasks. Extensive experiments on two LCR datasets show the superiority of GEAR over state-of-the-art methods while maintaining competitive judgment prediction performance. Moreover, we validate its robustness across languages and domains on a French SAR dataset.

ACKNOWLEDGMENTS

This work was funded by the National Key R&D Program of China (2023YFA1008704), the National Natural Science Foundation of China (No. 62376275, 62377044), Beijing Key Laboratory of Big Data Management and Analysis Methods, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, funds for building world-class universities (disciplines) of Renmin University of China, and PCC@RUC.

REFERENCES

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2017. Guided Open Vocabulary Image Captioning with Constrained Beam Search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 936–945.

[2] Joris Baan, Maartje ter Hoeve, Marlies van der Wees, Anne Schuth, and Maarten de Rijke. 2019. Do transformer attention heads provide transparency in abstractive summarization? *arXiv preprint arXiv:1907.00570* (2019).

[3] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150* (2020).

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

[5] Paheli Bhattacharya, Kripabandhu Ghosh, Arindam Pal, and Saptarshi Ghosh. 2020. *Hier-SPCNet: A Legal Statute Hierarchy-Based Heterogeneous Network for Computing Legal Case Document Similarity.* Association for Computing Machinery, New York, NY, USA, 1657–1660.

[6] Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive Entity Retrieval. In *International Conference on Learning Representations*. https://openreview.net/forum?id=5k8F6UU39V

[7] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. *arXiv preprint arXiv:2010.02559* (2020).

[8] Minmin Chen, Bo Chang, Can Xu, and Ed H Chi. 2021. User response models to improve a reinforce recommender system. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 121–129.

[9] Jiaxi Cui, Zongjian Li, Yang Yan, Bohua Chen, and Li Yuan. 2023. ChatLaw: Open-Source Legal Large Language Model with Integrated External Knowledge Bases. arXiv:2306.16092 [cs.CL]

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

[11] J. Edward Hu, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, and Benjamin Van Durme. 2019. Improved Lexically Constrained Decoding for Translation and Monolingual Rewriting. 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)*. Association for Computational Linguistics, Minneapolis, Minnesota, 839–850. https://doi.org/10.18653/v1/N19-1090

[12] Buomsoo Kim, Jinsoo Park, and Jihae Suh. 2020. Transparency and accountability in AI decision support: Explaining and visualizing convolutional neural networks for text information. *Decision Support Systems* 134 (2020), 113302. https://doi.org/10.1016/j.dss.2020.113302

[13] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[14] Michel CA Klein, Wouter Van Steenbergen, Elisabeth M Uijttenbroek, Arno R Lodder, and Frank van Harmelen. 2006. Thesaurus-based Retrieval of Case Law. *Frontiers in Artificial Intelligence and Applications* 152 (2006), 61.

[15] Haitao Li, Qingyao Ai, Jia Chen, Qian Dong, Yueyue Wu, Yiqun Liu, Chong Chen, and Qi Tian. 2023. SAILER: Structure-aware Pre-trained Language Model for Legal Case Retrieval. *arXiv preprint arXiv:2304.11370* (2023).

[16] Haitao Li, Yunqiu Shao, Yueyue Wu, Qingyao Ai, Yixiao Ma, and Yiqun Liu. 2023. LeCaRDv2: A Large-Scale Chinese Legal Case Retrieval Dataset. *arXiv preprint arXiv:2310.17609* (2023).

[17] Shuchang Liu, Qingpeng Cai, Bowen Sun, Yuhao Wang, Ji Jiang, Dong Zheng, Peng Jiang, Kun Gai, Xiangyu Zhao, and Yongfeng Zhang. 2023. Exploration and Regularization of the Latent Action Space in Recommendation. In *Proceedings of the ACM Web Conference 2023*. 833–844.

[18] Yifei Liu, Yiquan Wu, Yating Zhang, Changlong Sun, Weiming Lu, Fei Wu, and Kun Kuang. 2023. ML-LJP: Multi-Law Aware Legal Judgment Prediction. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Taipei, Taiwan) *(SIGIR '23)*. Association for Computing Machinery, New York, NY, USA, 1023–1034. https://doi.org/10.1145/3539618.3591731

[19] Antoine Louis and Gerasimos Spanakis. 2021. A statutory article retrieval dataset in French. *arXiv preprint arXiv:2108.11792* (2021).

[20] Antoine Louis, Gijs van Dijck, and Gerasimos Spanakis. 2023. Finding the Law: Enhancing Statutory Article Retrieval via Graph Neural Networks. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Dubrovnik, Croatia, 2753–2768. https://aclanthology.org/2023.eacl-main.203/

[21] 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*. 2342–2348.

[22] Akshay Minocha, Navjyoti Singh, and Arjit Srivastava. 2015. Finding Relevant Indian Judgments Using Dispersion of Citation Network. In *Proceedings of the 24th International Conference on World Wide Web*. 1085–1088.

[23] Marie-Francine Moens. 2001. Innovative techniques for legal text retrieval. *Artificial Intelligence and Law* 9 (2001), 29–57.

[24] Alexandra Olteanu, Jean Garcia-Gathright, Maarten de Rijke, Michael D Ekstrand, Adam Roegiest, Aldo Lipani, Alex Beutel, Alexandra Olteanu, Ana Lucic, Ana-Andreea Stoica, et al. 2021. FACTS-IR: fairness, accountability, confidentiality, transparency, and safety in information retrieval. In *ACM SIGIR Forum*, Vol. 53. ACM New York, NY, USA, 20–43.

[25] Matt Post and David Vilar. 2018. Fast Lexically Constrained Decoding with Dynamic Beam Allocation for Neural Machine Translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 1314–1324.

[26] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.

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

[28] Yunqiu Shao, Haitao Li, Yueyue Wu, Yiqun Liu, Qingyao Ai, Jiaxin Mao, Yixiao Ma, and Shaoping Ma. 2023. An Intent Taxonomy of Legal Case Retrieval. *ACM Transactions on Information Systems* (2023).

[29] 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*. 3501–3507.

[30] Yunqiu Shao, Yueyue Wu, Yiqun Liu, Jiaxin Mao, and Shaoping Ma. 2023. Understanding Relevance Judgments in Legal Case Retrieval. *ACM Transactions on Information Systems* 41, 3 (2023), 1–32.

[31] Zhongxiang Sun, Jun Xu, Xiao Zhang, Zhenhua Dong, and Ji-Rong Wen. 2022. Law Article-Enhanced Legal Case Matching: a Model-Agnostic Causal Learning Approach. *arXiv preprint arXiv:2210.11012* (2022).

[32] Zhongxiang Sun, Weijie Yu, Zihua Si, Jun Xu, Zhenhua Dong, Xu Chen, Hongteng Xu, and Ji-Rong Wen. 2023. Explainable Legal Case Matching via Graph Optimal Transport. *IEEE Transactions on Knowledge and Data Engineering* (2023).

[33] Zhongxiang Sun, Kepu Zhang, Weijie Yu, Haoyu Wang, and Jun Xu. 2024. Logic Rules as Explanations for Legal Case Retrieval. *arXiv preprint arXiv:2403.01457* (2024).

[34] Yi Tay, Vinh Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, Tal Schuster, William W Cohen, and Donald Metzler. 2022. Transformer Memory as a Differentiable Search Index. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 21831–21843. https://proceedings.neurips.cc/paper_files/paper/2022/file/892840a6123b5ec99ebaab8be1530fba-Paper-Conference.pdf

[35] Yujing Wang, Yingyan Hou, Haonan Wang, Ziming Miao, Shibin Wu, Qi Chen, Yuqing Xia, Chengmin Chi, Guoshuai Zhao, Zheng Liu, et al. 2022. A neural corpus indexer for document retrieval. *Advances in Neural Information Processing Systems* 35 (2022), 25600–25614.

[36] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8 (1992), 229–256.

[37] 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 (2021), 79–84.

[38] 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 (2021), 79–84.

[39] Xin Xin, Alexandros Karatzoglou, Ioannis Arapakis, and Joemon M Jose. 2020. Self-supervised reinforcement learning for recommender systems. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 931–940.

[40] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808* (2020).

[41] Weijie Yu, Liang Pang, Jun Xu, Bing Su, Zhenhua Dong, and Ji-Rong Wen. 2022. Optimal Partial Transport Based Sentence Selection for Long-form Document Matching. In *Proceedings of the 29th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 2363–2373. https://aclanthology.org/2022.coling-1.208

[42] 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 *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 657–668.

[43] Yiming Zeng, Ruili Wang, John Zeleznikow, and Elizabeth 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 9*. Springer, 339–345.

[44] ChengXiang Zhai et al. 2008. Statistical language models for information retrieval a critical review. *Foundations and Trends® in Information Retrieval* 2, 3 (2008), 137–213.

[45] Han Zhang and Zhicheng Dou. 2023. Case Retrieval for Legal Judgment Prediction in Legal Artificial Intelligence. In *China National Conference on Chinese Computational Linguistics*. Springer, 434–448.

[46] Xinyan Zhao and VG Vinod Vydiswaran. 2021. Lirex: Augmenting language inference with relevant explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 14532–14539.

[47] Haoxi Zhong, Zhengyan Zhang, Zhiyuan Liu, and Maosong Sun. 2019. *Open Chinese Language Pre-trained Model Zoo*. Technical Report. https://github.com/thunlp/openclap

[48] Yujia Zhou, Jing Yao, Zhicheng Dou, Ledell Wu, Peitian Zhang, and Ji-Rong Wen. 2022. Ultron: An Ultimate Retriever on Corpus with a Model-based Indexer. arXiv:2208.09257 [cs.IR]

[49] Shengyao Zhuang, Houxing Ren, Linjun Shou, Jian Pei, Ming Gong, Guido Zuccon, and Daxin Jiang. 2022. Bridging the gap between indexing and retrieval for differentiable search index with query generation. *arXiv preprint arXiv:2206.10128* (2022).