Leveraging Event Schema to Ask Clarifying Questions for Conversational Legal Case Retrieval

Bulou Liu
DCST, Tsinghua University
Quan Cheng Laboratory
Institute for Internet Judiciary,
Tsinghua University
lbl20@mails.tsinghua.edu.cn

Yiran Hu
Law School, Tsinghua University
huyr21@mails.tsinghua.edu.cn

Qingyao Ai
DCST, Tsinghua University
Quan Cheng Laboratory
Institute for Internet Judiciary,
Tsinghua University
aiqingyao@gmail.com

Yiqun Liu*
DCST, Tsinghua University
Quan Cheng Laboratory
Institute for Internet Judiciary,
Tsinghua University
yiqunliu@tsinghua.edu.cn

Yueyue Wu
DCST, Tsinghua University
Quan Cheng Laboratory
Institute for Internet Judiciary,
Tsinghua University
wuyueyue@mail.tsinghua.edu.cn

Chenliang Li
School of Cyber Science and
Engineering,
Wuhan University
cllee@whu.edu.cn

Weixing Shen
Law School, Tsinghua University
wxshen@mail.tsinghua.edu.cn

ABSTRACT

Legal case retrieval is a special IR task aiming to retrieve supporting cases for a given query case. Existing works have shown that conversational search paradigm can improve users' search experience in legal case retrieval. One of the keys to a practical conversational search system is how to ask high-quality clarifying questions to initiate conversations with users and understand their search intents. Recently, Large Language Models, such as ChatGPT and GPT-4, have shown superior ability in both open-domain QA and conversations with human. Thus it is natural to believe that they could be applied to legal conversational search as well. However, our preliminary study has shown that generating clarifying questions in legal conversational search with SOTA LLMs (e.g., GPT-4) often suffers from several problems such as duplication and low-utility contents. To address these problems, we propose LeClari, which leverages legal event schema as external knowledge to instruct LLMs to generate effective clarifying questions for legal conversational search. LeClari is constructed with a prompt module and a novel legal event selection module. The former defines a prompt with legal events for clarifying question generation and the latter selects potential event types by modeling the relationships of legal event types, conversational context, and candidate cases. We also propose ranking-oriented rewards and employ the reward augmented maximum likelihood (RAML) method to optimize LeClari directly based on the final retrieval performance of the conversational legal search system. Empirical results over two widely adopted legal case retrieval datasets demonstrate the effectiveness of our approach as compared with the state-of-the-art baselines.

CCS CONCEPTS

• **Information systems** → *Collaborative search.*

KEYWORDS

Conversational search, legal case retrieval, clarifying question

ACM Reference Format:
Bulou Liu, Yiran Hu, Qingyao Ai, Yiqun Liu*, Yueyue Wu, Chenliang Li, and Weixing Shen. 2023. Leveraging Event Schema to Ask Clarifying Questions for Conversational Legal Case Retrieval. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3583780.3614953

1 INTRODUCTION

In recent years, legal case retrieval has attracted much attention in the IR research community. It aims to retrieve supporting cases for a given query case and constitutes an essential component of a legal information system. In practice, prior cases are primary legal materials in various law systems. When using a conventional legal case retrieval system, users need to issue queries to express their information needs [8, 21], which could be complex and difficult to verbalize [31]. Conversational search is a rising topic in IR [26] and it can help users better express their information needs [4, 26], especially for complex search tasks [3, 32]. In legal case retrieval, studies have shown that conversational search paradigm can improve legal case retrieval from a variety of perspectives including but not limited to query formulation, users' satisfaction and search success [15, 17, 18].

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

© 2023 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-0124-5/23/10.
https://doi.org/10.1145/3583780.3614953

*Corresponding author

One of the key research problems in conversational search systems is how to ask good clarifying questions based on the conversation context so that we could better understand user intents and guide future conversations based on the user's answers. Recently, revolutionary Large Language Models (LLMs) techniques, such as ChatGPT [24] and GPT-4 [25], have shown strong zero-shot and few-shot generalization ability in many natural language processing tasks. Intuitively, it seems natural to apply LLMs to generate clarifying questions for legal conversational search, e.g., using simple prompts such as "based on the above conversation, please ask a clarifying question to further understand the background information of the legal case?". However, our preliminary study on the state-of-the-art LLMs (e.g., ChatGPT and GPT-4) has shown several problems that limit their performance in asking high-quality clarifying questions for legal conversational search. First, they sometimes generate clarifying questions that focus on facts that have already been presented in the previous context, which could provide low or no additional information to the conversation. Second, existing LLMs often ask general questions that are not relevant from the legal perspectives and thus provide limited benefits for the performance of downstream legal retrieval models. Because LLMs are usually built with open-domain data and are not trained specifically for clarifying question generation, they do not know what to ask and how to ask effective questions in legal case retrieval.

Inspired by recent studies on constrained question generation [36] and conversational product search [6, 43, 45], we propose LeClari, a conversational search model that generates high-quality clarifying questions for conversational legal case retrieval. LeClari is constructed with a prompt module and an event selection module. The event selection module iteratively selects event types from a legal event schema to guide LLMs to ask clarifying questions with the prompt module. The legal event schema can be considered as a special kind of legal database that contains multiple types of legal events with their descriptions. Here we leverage the existing legal event schema LEVEN [38] for the prompt construction in LeClari. The event schema LEVEN reasonably divides the key facts in criminal cases into 108 event types and can be utilized as external knowledge to promote downstream legal applications. By selecting event types from LEVEN, LeClari can ask questions effectively to narrow down the search space of downstream retrieval models and thus improve the performance of the whole system.

However, adopting existing conversational models to the selection of legal event types is suboptimal because they mostly ignore the connections between event type selection and downstream retrieval tasks. To this end, LeClari explicitly models the relationships of legal event types, conversational context, and potential candidate cases retrieved by downstream legal case retrieval models together in its event selection module. Further, we propose ranking-oriented rewards and employ the reward augmented maximum likelihood (RAML) method [22] to optimize LeClari directly for downstream retrieval metrics such as MAP and NDCG.

We conduct empirical experiments on two widely adopted legal case retrieval datasets, including the LeCaRD [20] dataset and the CAIL2022-LCR dataset. For evaluation, we compare with several other event type selection strategies in conversational models to verify the effectiveness of our model. Empirical results demonstrate that our model can select appropriate event types for LLMs to construct useful clarifying questions and improve the legal case retrieval performances significantly than all the baselines on the two datasets.

2 RELATED WORK

2.1 Legal case retrieval

Legal case retrieval is a specialized IR task [33, 34, 39]. Several approaches have been explored in previous research of legal IR, including knowledge engineering-based techniques and NLP-based methods [7]. For instance, [27] combined symbolic and connectionist artificial intelligence techniques to integrate both symbolic and sub-symbolic information in legal domain. [28] developed a legal knowledge-based framework to overcome synonymy and ambivalence of words in query process and enhance the user's query for retrieving truly relevant legal judgments. However, these existing legal case retrieval systems still followed a traditional search paradigm in which users issued keyword-based queries to describe their information needs [8, 21]. With the rapid development of deep learning, applying pre-trained language models (PLMs) to legal case retrieval has received a huge success. [30] proposed a BERT-based neural network to model paragraph-level interactions for legal case retrieval. And [19, 29, 30] suggested that BERT-based neural networks improved the performance of the legal case retrieval task significantly. In recent years, knowledge transfer has become a popular research topic [5, 14, 16]. In addition to the original PLMs pre-trained by multiple resources, [10–13] demonstrated that domain-adaptive pretraining improves the performance of PLMs in domain-specific tasks. For example, [42] and [37] used large legal corpus to pre-train BERT and Longformer, respectively. Both models outperform their precedent PLMs in legal tasks.

2.2 Learning to Ask Clarifying Questions

Clarifying questions are important parts to improve conversational search systems and has attracted much attention in the IR research community [1, 40]. Recently, LLMs [24, 25] are revolutionizing natural language processing and have great potential in promoting clarifying question generation. As LLMs are not specifically trained for clarifying question generation, researchers relied on providing constrained prompts to elicit their zero-shot generalization ability. In open-domain conversational search, [36] proposed to constrain the clarifying question decoding with search facets to solve the cold-start problem. And in conversational product search, researchers focused on asking clarifying questions based on a pre-defined product aspect set and propose a series of learning to ask strategies. [44] learned to ask a good question based on user preferences and the rewards over question performances. [41] predicted the next question to ask to the user by maximizing the probability of the next question based on the softmax output layer for probability estimation. And [43] proposed a set of systematic learning to ask strategies, including both greedy (GBS) and explore-exploit (bandit learning) strategies. Compared to these studies, we propose LeClari which selects event types from the legal event schema to generate clarifying questions and promote the legal case retrieval systems.

3 PRELIMINARY STUDY

In this section, we conduct a preliminary study to investigate the clarifying question generation performances of LLMs in legal case

Table 1: The Quality Scores of Clarifying Questions by LLMs

Aspect	LeCaRD		CAIL2022-LCR	
	ChatGPT	GPT-4	ChatGPT	GPT-4
Relevance	0.9907	0.9953	0.9885	0.9923
Answerability	0.6075	0.6636	0.6308	0.6731
Information Gain	0.3645	0.3738	0.3846	0.3954

retrieval scenario, which leads to the following two research questions:

- **RQ1:** What are the qualities of the clarifying questions generated by LLMs in legal scenarios, independent of the search system?
- **RQ2:** Can the clarifying questions generated by LLMs provide benefits for the performance of downstream legal retrieval models?

3.1 Conversation Construction

To address the above two research questions, we utilize two widely adopted legal case retrieval datasets LeCaRD [20] and CAIL2022-LCR[1] to construct the conversations and evaluate the clarifying questions. LeCaRD is the first criminal case retrieval dataset under the Chinese law system. Challenge of AI in Law (CAIL) is a competition held annually under the guidance of the Supreme People's Court and the Chinese Information Processing Society of China to promote AI technology and a higher level of digital justice since 2018. As one of the eight tasks in CAIL 2022, the legal case retrieval task provides a dataset named CAIL2022-LCR. These two datasets contain several query cases (i.e., the complete fact description parts of case documents) and each query case corresponds to a candidate case pool with a size of 100. It is required to select relevant cases from the candidate case pool for each query case. Every candidate case has a four-level relevance label annotated by criminal law experts. To analyze the LLMs' abilities to generate legal clarifying questions, we construct conversations based on the two datasets following the steps below.

(1) *Initial query construction.* We invited a PhD student major in criminal law to select 1-2 sentences from each query case and rewrite them into coherent statements as the initial query. Note that we focus on situations where the search system needs to ask clarifying questions to further understand the background information of the query case. Therefore, we asked the PhD student to leave some important information from the query case when formulating the initial query and mark it in the query case. We hope that the clarifying questions can help the search system to complete this part of information. We utilize the initial query as the start of the conversation.

(2) *Clarifying question generation.* Then we apply LLMs including ChatGPT and GPT-4 to generate a clarifying question as the system reply based on the contextual conversation information. Specifically, we incorporate the conversation into the following prompt and feed the prompt into LLMs to generate the clarifying question:

> *You are now a knowledgeable judge in law. The current conversation between you and the user is as follows: [the conversation]. Based on the above conversation, what clarifying question can you ask to further understand the background information of the case?*

(3) *User response simulation.* After the system asks the clarifying question, the user needs to answer it as the response based on the query case. This process is similar to reading comprehension (i.e., answer the question based on the background information). Considering that the cost of collecting the user feedback is prohibitive, we use another LLM as a user simulator to generate the answer instead. Specifically, we incorporate complete fact description part of the original case document (i.e., the query case) from which we generate the initial query and the last clarifying question into the following prompt and feed the prompt into the LLM to generate the answer:

> *Please read the following background information: [the original fact description of case document used to generate the query]. And answer the following question: [clarifying question].*

Based on previous user study results [18], the average number of clarifying questions is 4 in the scenario of conversational legal case retrieval. Therefore, for each query case, we generate four clarifying questions to complete the conversation construction process.

In addition, to ensure the quality of the user simulator, we randomly selected 20 conversations from two datasets (a total of 40 conversations, 160 clarifying questions and answers) and invited another three graduate students major in criminal law to annotate the answers to each clarifying question (1 point - correct, 0 point - incorrect). The Fleiss's κ among three assessors was 0.913, indicating almost perfect agreement [9]. If there were disagreements, we took the result of the majority vote. We find that only 3 answers are annotated as incorrect. This shows that the user responses generated by the LLM is convincing.

3.2 Clarifying Question Quality Analysis

To address **RQ1**, we evaluate the qualities of the clarifying questions generated by the LLMs from multiple aspects, independent of the search system. Specifically, the three graduate students labeled each clarifying question in the constructed conversations according to topic relevance, answerability, and information gain. They denote whether the clarifying question is relevant to the initial query, can be answered based on the query case and provides additional information, respectively. Each labeling task asks the annotators to assign a label to the clarifying question (1 point - relevant/answerable/provide additional information, 0 point - irrelevant/unanswerable/not provide additional information). The Fleiss's κ scores of the three tasks were 0.863, 0.834 and 0.818, respectively, indicating almost perfect agreement. We obtain the labels based on majority voting.

We calculate the average quality scores for these three aspects, respectively. The results are shown in Table 1. We find that almost all clarifying questions are relevant to the initial query. However, more than 30% of clarifying questions can not be answered, indicating that the corresponding information is not mentioned in the

---

[1] http://cail.cipsc.org.cn/index.html

Table 2: Retrieval performance comparison of the conversations with/without clarifying questions in terms of NDCG@10. There are no significant differences between the performances with or without clarifying questions.

Dataset	BERT-Crime			LawFormer		
	w/o clarify	ChatGPT	GPT-4	w/o clarify	ChatGPT	GPT-4
LeCaRD	0.5369	0.5447	0.5481	0.5021	0.5086	0.5099
CAIL-LCR	0.5999	0.6085	0.6105	0.5569	0.5591	0.5603

query case. In addition, only 35-40% of clarifying questions can help the search system obtain additional information. This means for 20-30% of clarifying questions, although their corresponding answers can be found in the query case, the content they ask about has already been presented in the existing conversation. But LLMs fail to realize this.

3.3 Retrieval Performance Analysis

To address **RQ2**, we investigate whether the clarifying questions generated by LLMs can obtain useful information to improve the retrieval performance in legal scenarios. Specifically, we first fine-tune two legal pre-train language models, BERT-Crime [42] and LawFormer [37], on a conversational legal case retrieval dataset [18] to enhance their conversational search abilities. We use pair-wise loss to train them by feeding the concatenation of the conversation and the candidate case document. We hope that they can find relevant cases when the information is sufficient, so as to determine whether the clarifying questions have obtained useful information based on retrieval performance. Then we utilize them to compute the relevance scores between the constructed conversations and candidate cases. Finally, we compare the retrieval performances between using the conversations without clarifying questions (i.e., only the initial query) and those with clarifying questions generated by LLMs.

We use NDCG@10 as the retrieval metric and the results are shown in Table 2. The differences are all not significant at 0.05 level with a two-tailed pairwise t-test. We find that the clarifying questions proposed by LLMs (i.e., *"ChatGPT"* and *"GPT-4"* group) do not significantly improve legal case retrieval performance, although the retrieval metrics are slightly worse when clarifying questions were not proposed (i.e., *"w/o clarify"* group). This indicates that although some clarifying questions by LLMs can obtain additional background information, they do not help improve the conversational legal case retrieval performances.

3.4 Summary

Regarding the two research questions, we find two disadvantages of legal clarifying questions which are generated by LLMs directly: (1) As for **RQ1**, although almost all the clarifying questions are relevant to the search task, more than half of them are unable to obtain additional information. Especially some of them focus on facts that have already been presented in the previous context, which could provide low or no additional information to the conversation. (2) As for **RQ2**, they are not relevant from the legal perspectives and thus provide limited benefits for the performance of downstream legal retrieval models.

Table 3: Statistics and examples of event schema LEVEN

Category	#Type	Type Examples
General_behaviors	40	Buying, Selling
Prohibited_acts	40	Bodily_harm, Escaping
Consequences	7	Death, Injury
Judicature_related	13	Surrendering
Accident	4	Traffic_accident
Natural_disaster	4	Flood_and_waterlogging

4 LECLARI

In this section, we present LeClari which is a conversational search model that generates high-quality legal clarifying questions (ref. Figure 1). We first introduce the Prompt Module (PM) with legal event types to generate clarifying questions by LLMs and the workflow to interact with users. Then we show the Event Selection Module that selects appropriate event types for Prompt Module and the model training strategy.

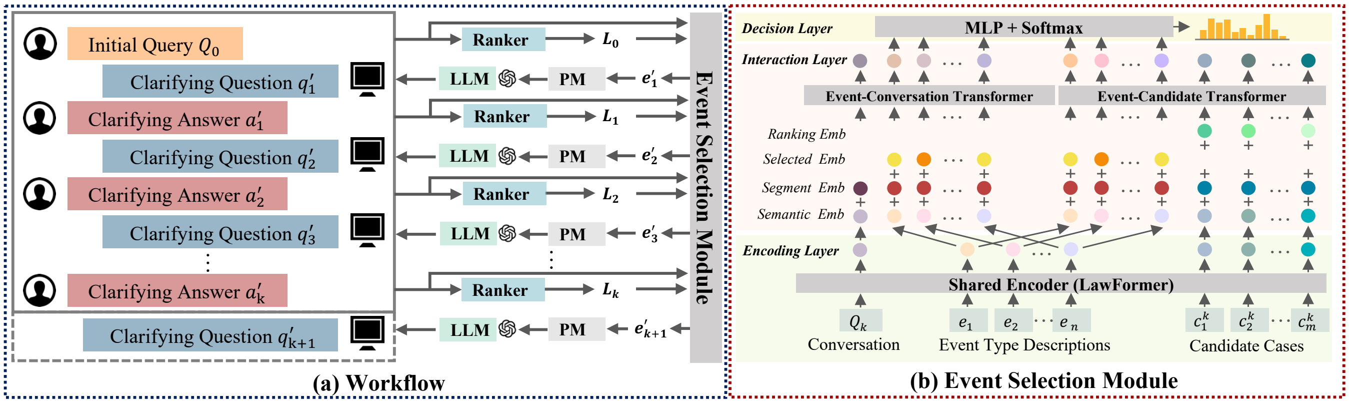
4.1 Prompt Module

To overcome the two disadvantages mentioned in Section 3 and generate useful clarifying questions for legal case retrieval, we propose the Prompt Module by leveraging the legal event schema LEVEN[2] [38] for LLMs prompt constructions. LEVEN is constructed based on the law articles, legal textbooks and case documents. It can be considered as a special kind of legal database which groups the facts of criminal law into 108 event types. The event types can be divided into 6 categories and the examples are shown in Table 3. The first three categories are related to various human behaviors. The fourth and fifth are related to results. The last one is related to majeure. Each event type has a textual description, like "*Escaping: Fleeing and hiding to avoid unfavorable circumstances.*" Therefore, the event types can be added as constrains into the LLMs prompts to generate clarifying questions. The constrained generated questions can help the search systems know the detailed facts in the background information related to the event type. On the one hand, we can force to use different event types as the constrains in different rounds. It can force LLMs to generate different clarifying questions and avoid asking about the same contents. On the other hand, [38] found that the legal case retrieval performance could be improved when they focused on the facts related to event types of LEVEN in the query case. Intuitively, selecting appropriate event types from LEVEN can offset the disadvantages mentioned in Section 3. So at each round, the Prompt Module selects an event type and incorporates it into a pre-defined LLMs prompt. The descriptions are also used to explain the event type. For example, when the user issues an initial query "Alice drove a car at night and crashed into Bob", we can select "Escaping" event type to construct the following prompt:

> **Prompt**: *You are now a knowledgeable judge in law. The current conversation between you and the user is as follows: User: Alice drove a car at night and crashed into*

---

[2] https://github.com/thunlp/LEVEN



(a) Workflow (b) Event Selection Module

Figure 1: The overview of LeClari. PM denotes the Prompt Module.

> *Bob. Based on the above conversation, what "Escaping"-related clarifying question can you ask to further understand the background information of the case? Here "Escaping" means fleeing and hiding to avoid unfavorable circumstances.*

And ChatGPT can generate the following clarifying questions based on this prompt, which is highly related to the event type: "*Can you provide any information on whether Alice attempted to flee or hide after the accident?*"

4.2 Workflow

After defining the Prompt Module, we can formalize the workflow of LeClari to interact with users based on it (shown in Figure 1(a)). Assume that we have N event types $E = \{e_1, e_2, ..., e_N\}$ and M candidate cases $C = \{c_1, c_2, ..., c_M\}$. We now consider a conversation search scenario where users and the search system are discussing and finding relevant legal case documents for a specific query case. A user with a legal query case issued an initial query Q_0 in the form of natural language to start a search session. During the session, the search system selects an event type e' (by the Event Selection Module in Section 4.3) for the Prompt Module to generate a clarifying question q'. And the user provides an associated answer a' for the clarifying question based on her query case. In the conversation for each search session, the system asks the user a sequence of clarifying questions and collects a sequence of user answers. Suppose LeClari asks K-round clarifying questions, the sequence S_K of the events, clarifying questions and answers are represented as follows:

$$S_K = \{(e'_1, q'_1, a'_1), (e'_2, q'_2, a'_2), ..., (e'_K, q'_K, a'_K)\} \tag{1}$$

Based on the sequence S_K, the current conversation Q_K can be denoted as follows:

$$Q_K = \{Q_0, (q'_1, a'_1), (q'_2, a'_2), ..., (q'_K, a'_K)\} \tag{2}$$

Finally, we apply the fine-tuned BERT-Crime or LawFormer in Section 3.3 as the Ranker. They rank the candidate cases C based on the conversation Q_K and obtain the ranking list L_K:

$$L_K = Ranker(Q_K, C) = \{c^K_1, c^K_2, ..., c^K_M\} \tag{3}$$

where L_K is a permutation of the candidate case set C. Thus, the sequence of actions in the conversation can be represented as:

$$\begin{aligned} Q_0 \rightarrow L_0 \rightarrow e'_1, q'_1, a'_1, Q_1 \rightarrow L_1 \rightarrow e'_2, q'_2, a'_2, Q_2 \rightarrow L_2 \rightarrow \\ ... \rightarrow e'_K, q'_K, a'_K, Q_K \rightarrow L_K \end{aligned} \tag{4}$$

The goal of the search system is to maximize the retrieval metrics of the ranking list L_K. We compute the retrieval metrics each round until asking K clarifying questions. Note that K is a pre-defined number and we leave the selection of the number of clarifying questions as future work.

4.3 Event Selection Module

The relevance estimation in legal case retrieval has already achieved good solutions through pre-trained models. So we fix the parameters of Ranker and focus on selecting appropriate event types and asking useful clarifying questions in this paper. We design the Event Selection Module (ref. Figure 1(b)), which selects the $(k+1)$-round event types e'_{k+1} for the Prompt Module given the initial query Q_0, the k-round sequence S_k and the k-round ranking list L_k. It contains an encoding layer, an interaction layer and a decision layer.

4.3.1 *Encoding Layer.*

To mine the rich semantic information in the conversations, event descriptions and candidate cases, we apply a shared encoding layer to generate semantic embeddings for them.

As for the conversation Q_k, we set LawFormer [37] as our encoder, which is a Longformer-based pre-trained language model for legal long documents understanding. In detail, the conversation Q_k contains the initial query and a sequence of clarifying questions and answers. It can be denoted as $\{Q_0, (q'_1, a'_1), ..., (q'_k, a'_k)\}$. Usually, a specific token [CLS] is inserted as the first token and another token [SEP] is utilized to split different segments. Therefore, the semantic embeddings of the conversations can be obtained as follows:

$$x^Q = Encoder([CLS] \circ Q_0 \circ [SEP] \circ q'_1 \circ a'_1 \circ ... \circ [SEP] \circ q'_k \circ a'_k) \tag{5}$$

Here, $\circ$ denotes the concatenation of two sequences. And we use the [CLS] representations as the semantic embeddings of conversations.

In addition, given all event types $E = \{e_1, e_2, ..., e_N\}$, we also apply LawFormer to map each event type to a dense representation based on its description. Specifically, the description of i-th event type e_i is a word sequence $(w^e_{i1}, w^e_{i2}, ..., w^e_{iV})$. V represents the maximum length of event description. We also insert the [CLS] token as the first token and the semantic embedding of i-th event type is generated as follows:

$$x_i^e = Encoder([CLS], w_{i1}^e, w_{i2}^e, ..., w_{iV}^e) \tag{6}$$

We also use the [CLS] representations as the semantic embeddings of event types.

As for the ranking list of candidate cases $C = \{c_1^k, c_2^k, ..., c_M^k\}$, the j-th ranked candidate case can be denoted as a word sequence $(w_{j1}^c, w_{j2}^c, ..., w_{jU}^c)$ and U represents the maximum length of candidate case. Similarly, we encode the j-th ranked candidate case by LawFormer and use the [CLS] representation as its semantic embedding:

$$x_j^c = Encoder([CLS], w_{j1}^c, w_{j2}^c, ..., w_{jU}^c) \tag{7}$$

4.3.2 Interaction Layer. The final aim of LeClari is to find relevant legal cases. So the Event Selection Module needs to consider about not only the conversation context but also the ranking list of candidate cases. Therefore, we apply an interaction layer which aims to enhance the representations of the event types based on the conversation and candidate cases. To achieve this purpose, we leverage two Vanilla Transformers to represent event embeddings. One of them (named Event-Conversation Transformer) encodes the event types with conversations and another (named Event-Candidate Transformer) encodes the event types with the ranking lists of candidate cases. The multi-head attention mechanism used in the Transformer captures interaction information between the event types and the conversations/candidate cases.

As the input of the Event-Conversation Transformer, we utilize the semantic embeddings $\{x^Q, x_1^e, x_2^e, ..., x_N^e\}$ obtained in Section 4.3.1 and add the segment embeddings which help the model distinguish between conversations and event types (denoted as $\mathbf{I}_{con}^{seg}$ and $\mathbf{I}_{event}^{seg}$, respectively). Specifically, the input embedding of the conversation is represented as follows:

$$\mathbf{I}^Q = x^Q + \mathbf{I}_{con}^{seg} \tag{8}$$

Meanwhile, because our task aims to select the next round event type for clarifying question generation, the event selection history should be taken into consideration. For example, when the conversation contains the event "Robbery", it is necessary to select "Injury" to construct a clarifying question to know whether the defendant has caused injuries. Therefore, given the selected event types in the previous k round $E_k' = \{e_1', e_2', ..., e_k'\}$, the input embeddings of event types add selected embeddings (denoted as $\mathbf{I}_{\mathbb{S}(e_i \in E_k')}^{sel}$). When e_i has been selected, $e_i \in E_k'$ is true and $\mathbb{S}(e_i \in E_k') = 1$, otherwise $e_i \in E_k'$ is false and $\mathbb{S}(e_i \in E_k') = 0$. Specifically, the input embedding of the i-th event type is represented as follows:

$$\mathbf{I}_i^e = x_i^e + \mathbf{I}_{event}^{seg} + \mathbf{I}_{\mathbb{S}(e_i \in E_k')}^{sel} \tag{9}$$

Based on the input embeddings of the conversation and event types, the Event-Conversation Transformer can generate the enhanced representations as follows:

$$[h^Q, h_1^{eQ}, ..., h_N^{eQ}] = Transformer(\mathbf{I}^Q, \mathbf{I}_1^e, ..., \mathbf{I}_N^e) \tag{10}$$

where h^Q is the enhanced conversation representation and h_i^{eQ} is the i-th event type enhanced representation which is combined with conversational information.

As for the input of the Event-Candidate Transformer, we utilize the same event input embeddings as the Event-Conversation Transformer. And the input embedding of the j-th ranked candidate case also contains the semantic embedding x_j^c and segment embedding (denoted as $\mathbf{I}_{case}^{seg}$). In addition, because the final aim of our task is to improve the legal case retrieval performance(i.e., rerank the candidate list), the current ranking position information is important for event selection decision. Therefore, the ranking embedding $\mathbf{I}_j^{rank}$ for c_j^k is added to the input, which helps the model distinguish between candidate cases on different ranks:

$$\mathbf{I}_j^c = x_j^c + \mathbf{I}_{case}^{seg} + \mathbf{I}_j^{rank} \tag{11}$$

Then we can obtain the enhanced representations by the Event-Candidate Transformer as follows:

$$[h_1^{eC}, ..., h_N^{eC}, h_1^C, ..., h_M^C] = Transformer(\mathbf{I}_1^e, ..., \mathbf{I}_N^e, \mathbf{I}_1^c, ..., \mathbf{I}_M^c) \tag{12}$$

where h_i^{eC} is the i-th event type enhanced representation which is combined with the ranking list of candidate cases and h_j^C is the enhanced representation of the j-th ranked candidate case. In the two Transformers, the input embeddings are all randomly initialized and trainable, except the semantic embeddings.

4.3.3 Decision Layer. The two Vanilla Transformers output two kinds of the i-th event type representations h_i^{eQ} and h_i^{eC}, respectively. The former is combined with contextual conversations and the latter is combined with the candidate ranking lists. Here we need to generate a list of clarifying scores to select the next round event type. We import these representations into an MLP followed by a softmax layer to get the predictions:

$$s = \text{Softmax}(\text{MLP}([h_1^{eQ}, h_2^{eQ}, ..., h_N^{eQ}, h_1^{eC}, h_2^{eC}, ..., h_N^{eC}])) \tag{13}$$

Here $s = \{s_1, s_2, ..., s_N\}$ is a list of clarifying scores for the event types. Namely, s_i denotes the confidence for LeClari to select the i-th event type for clarifying question generation. Finally, we select the event type with the highest clarifying score among the event types which have never been selected in the previous rounds.

4.4 Model Training

As mentioned before, the key to improve LeClari is to select appropriate event types for the Prompt Module by the Event Selection Module. Here we introduce the training strategy to optimize the Event Selection Module, including the loss function and training samples.

4.4.1 Ranking-oriented Loss Function. We hope to enhance the ability of LeClari to generate high-quality clarifying questions to improve the legal case retrieval performance (i.e., retrieval metrics). The key is that the Event Selection Module can select appropriate event types for clarifying question generation. Therefore, we design the ranking-oriented rewards as the target clarifying scores for event types. Specifically, given the k-th round conversation Q_k, the event types E, the selected event types E_k' and the ranking list L_k, we can generate one round clarifying question and answer (q_{k+1}^i, a_{k+1}^i)

by LLMs for the i-th event type based on Q_k, following the event-based prompt in Section 4.1 and the user simulation prompt in Section 3. In this way, we can obtain $(k+1)$-round conversations for all event types (denoted as Q^i_{k+1} for the i-th event type e_i). Then we feed the i-th new conversation Q^i_{k+1} into the ranking models (i.e., BERT-Crime or LawFormer) to generate the new ranking list L^i_{k+1}. The ranking-oriented reward for the i-th event is computed as follows:

$$\gamma(i) = \begin{cases} 0 & e_i \in E'_k \\ max(\mathbf{M}(L^i_{k+1}) - \mathbf{M}(L_k), 0) & e_i \notin E'_k \end{cases} \tag{14}$$

where $\mathbf{M}(L)$ is the retrieval metric score (e.g., MAP and NDCG) of the ranking list L. So the rewards can reflect the usefulness of each event type for the current conversation Q_k.

Based on the rewards, we can use Maximum Likelihood Estimation (MLE) to optimize LeClari directly. Specifically, the event type with the highest reward (denoted as i^*) is defined as the ground truth and the loss function is as follows:

$$\mathcal{L}_{MLE} = -\sum_i \delta_{i^*}(i) \log s_i \tag{15}$$

where δ_{i^*} is the Dirac distribution of the ground-truth event type, i.e., $\delta_{i^*}(i^*) = 1$ and $\delta_{i^*}(i) = 0$ for other i.

As we can see, the MLE criterion ignores the structure of the output space by treating all the outputs that do not match the ground-truth as equally poor, and thus brings the discrepancy between training and test. So we propose to take into account the alternative outputs beyond the ground truth for better model learning. Specifically, we try to derive the new target distribution by employing Reward Augmented Maximum Likelihood (RAML) [22]. We normalize these target clarifying scores γ to obtain the distribution of the outputs by a softmax layer and replace the Dirac distribution in the loss function:

$$y_i = \frac{\exp(\gamma(i))}{\sum_{n=1}^{N} \exp(\gamma(n))}, \quad \mathcal{L}_{RAML} = -\sum_i y_i \log s_i \tag{16}$$

4.4.2 Training Conversations Sampling. Considering there are only the query cases with their candidate cases in the original legal case retrieval datasets, we need to construct training conversations to apply the ranking-oriented loss function. We utilize the initial queries in Section 3 and define an event sampling strategy to generate conversations automatically. Specifically, suppose there are G query cases in the dataset, we construct D conversations for each query case and each conversation contains $(K-1)$ round event-type related clarifying questions. K is the pre-defined maximum of the clarifying questions, so we can obtain $(G + G * D * (K-1))$ training samples (including G initial queries and $G * D$ conversations each round). In addition, to avoid that most of the clarifying questions by random sampling do not provide more useful information compared to the initial query, we use the target clarifying scores in Section 4.4.1 to define the dynamic probability of the i-th event type sampled in the next round based on the current conversation:

$$p_i = \frac{\exp(\lambda * \gamma(i))}{\sum_{n=1}^{N} \exp(\lambda * \gamma(n))} \tag{17}$$

where λ is a pre-defined parameter. When λ is higher, we tend to sample more useful events in the conversation. And if $\lambda = 0$, the sampling method degenerates into random sampling. Given the initial query, we sample event types round by round based on the dynamic sampling probability distributions until $(K-1)$ round clarifying questions and answers have been generated.

5 EXPERIMENTS

This section reports the experimental results. We first introduce the evaluation scheme and baseline models. Then we present overall performance comparison results, further analyses and a case study.

5.1 Evaluation Scheme

Evaluation Protocol. We aim to evaluate the clarifying questions for conversational legal case retrieval and use the same user simulation method in our preliminary study. Specifically, the conversation is started with an initial query and the user simulator generates the answer to the clarifying question by LLMs to continue the conversations. The user simulator is the same LLM for the clarifying question generation and its prompt has been shown in Section 3. We also generate 3 conversations for each query by each model to eliminate the effects of LLMs randomness. Until K clarifying questions have been generated (K is pre-defined), we apply pre-trained legal case retrieval models to obtain the final ranking lists. Here we utilize the LeCaRD and CAIL2022-LCR with the corresponding initial queries for evaluation, both following a 5-fold cross-validation. And we also apply the same LLMs (i.e., ChatGPT and GPT-4) and legal case retrieval models (i.e., BERT-Crime and LawFormer) in our preliminary study.

Metrics. We evaluate the performances of models using three metrics: Mean Average Precision (MAP), Precision@5 (P@5) and NDCG@10. Notably, we merge the four-level label in legal case retrieval datasets into binary when measuring MAP and P@5. Only cases with the highest relevance label are regarded as relevant cases and the rest are regarded as irrelevant.

5.2 Baselines

For a comprehensive evaluation, we compared our method with the following baselines. (1) **"w/o Clarify"** : does not generates clarifying questions and just uses the initial query for retrieval. (2) **"w/o Event"**: uses the prompt in our preliminary study to generate clarifying questions directly without incorporating the event type information. (3) Event selection models: only replaces the Event Selection Module in LeClari with other selection strategies.

As for the event selection models, we apply the following strategies to select event types. Previous work [43] has shown that they are effective in aspect selection for conversational product search. **Random** selects event types randomly and **MaxE** selects the most frequent event types in candidate cases. **GBS** [23] selects event types to best split the current ranking list corresponding to the candidate cases closest to two halves. **LinRel** [2] estimates a linear regression model and makes use of side information to estimate the relevance score of an event type. **GP+UCB/EI** [43] models the event type selection as a Gaussian Process and uses two acquisition functions (i.e., Upper Confidence Bound and Expected Improvement) to select them. Note that these strategies need to use one-hot vector (i.e., including the event type or not) to represent the query cases and the candidate cases. Considering that the human annotations are prohibitive, we train DMBERT [35], which is the best

Table 4: Performance Comparison on the LeCaRD dataset. The best results are highlighted with boldface. † denotes that LeClari performs significantly better than the baseline at 0.05 level with a two-tailed pairwise t-test.

Method	ChatGPT+BERT-Crime			ChatGPT+LawFormer			GPT4+BERT-Crime			GPT4+LawFormer		
	MAP	P@5	NDCG@10	MAP	P@5	NDCG@10	MAP	P@5	NDCG@10	MAP	P@5	NDCG@10
w/o Clarify	0.3227†	0.2130†	0.5369†	0.2768†	0.1780†	0.5021†	0.3227†	0.2130†	0.5369†	0.2768†	0.1780†	0.5021†
w/o Event	0.3262†	0.2170†	0.5447†	0.2811†	0.1820†	0.5086†	0.3284†	0.2160†	0.5481†	0.2820†	0.1830†	0.5099†
Random	0.3193†	0.2060†	0.5282†	0.2699†	0.1740†	0.5033†	0.3247†	0.2110†	0.5322†	0.2733†	0.1770†	0.5054†
MaxE	0.3435†	0.2310†	0.5518†	0.2942†	0.1970†	0.5244†	0.3485†	0.2330†	0.5574†	0.2976†	0.2020†	0.5289†
GBS	0.3689†	0.2220†	0.5701†	0.2894†	0.2110†	0.5438†	0.3745†	0.2260†	0.5744†	0.2939†	0.2150†	0.5466†
LinRel	0.3791†	0.2380†	0.5634†	0.2982†	0.2310†	0.5599†	0.3843†	0.2410†	0.5665†	0.3025†	0.2360†	0.5650†
GP+UCB	0.3975†	0.2630†	0.5982†	0.3264†	0.2390†	0.5742†	0.4009†	0.2690†	0.6012†	0.3296†	0.2410†	0.5791†
GP+EI	0.3821†	0.2540†	0.5721†	0.3220†	0.2390†	0.5583†	0.3846†	0.2590†	0.5749†	0.3264†	0.2420†	0.5607†
LeClari	**0.4226**	**0.2860**	**0.6217**	**0.3475**	**0.2640**	**0.5972**	**0.4278**	**0.2890**	**0.6249**	**0.3507**	**0.2690**	**0.6012**

Table 5: Performance Comparison on the CAIL2022-LCR dataset. The best results are highlighted with boldface. † denotes that LeClari performs significantly better than the baseline at 0.05 level with a two-tailed pairwise t-test.

Method	ChatGPT+BERT-Crime			ChatGPT+LawFormer			GPT4+BERT-Crime			GPT4+LawFormer		
	MAP	P@5	NDCG@10	MAP	P@5	NDCG@10	MAP	P@5	NDCG@10	MAP	P@5	NDCG@10
w/o Clarify	0.4078†	0.3230†	0.5999†	0.3703†	0.2590†	0.5569†	0.4078†	0.3230†	0.5999†	0.3703†	0.2590†	0.5569†
w/o Event	0.4157†	0.3310†	0.6085†	0.3795†	0.2630†	0.5591†	0.4136†	0.3290†	0.6105†	0.3784†	0.2720†	0.5603†
Random	0.4022†	0.3170†	0.5993†	0.3672†	0.2540†	0.5538†	0.4051†	0.3210†	0.6084†	0.3653†	0.2630†	0.5667†
MaxE	0.4239†	0.3360†	0.6148†	0.3914†	0.2760†	0.5741†	0.4209†	0.3470†	0.6384†	0.3842†	0.2880†	0.5856†
GBS	0.4537†	0.3310†	0.6340†	0.3889†	0.2900†	0.5918†	0.4516†	0.3420†	0.6590†	0.3801†	0.3020†	0.6020†
LinRel	0.4592†	0.3430†	0.6317†	0.4007†	0.3070†	0.6062†	0.4565†	0.3600†	0.6488†	0.3882†	0.3270†	0.6235†
GP+UCB	0.4776†	0.3680†	0.6648†	0.4328†	0.3200†	0.6244†	0.4775†	0.3860†	0.6881†	0.4185†	0.3290†	0.6367†
GP+EI	0.4622†	0.3640†	0.6365†	0.4322†	0.3170†	0.6046†	0.4587†	0.3730†	0.6584†	0.4166†	0.3260†	0.6179†
LeClari	**0.4993**	**0.3960**	**0.6830**	**0.4551**	**0.3470**	**0.6397**	**0.5025**	**0.4120**	**0.7104**	**0.4435**	**0.3550**	**0.6597**

legal event detection model, on LEVEN and apply it to annotate the event types for the query cases and candidate cases. On LeCaRD, the query cases and candidates contain 14.83 and 9.72 events on average, respectively. And on CAIL2022-LCR, the query cases and candidates contain 13.85 and 10.69 events on average, respectively.

5.3 Performance Comparison

Table 4 and 5 show the results in the two datasets. We can draw the following conclusions from the results. (1) The clarifying questions directly generated by LLMs do not improve the conversational legal case retrieval performance significantly like our findings in the preliminary study. (2) Almost all event type selection models (except "Random") can assist LLMs in generating clarifying questions which improve the performance of legal case retrieval. It indicates that the Prompt Module is useful, which leverages the event schema LEVEN and designs effective event-based LLMs prompts for legal clarifying question generation. (3) LeClari significantly performs better than all the baselines on the two datasets. It demonstrates that the Event Selection Module and the ranking-oriented training strategy are effective, which successfully connect the event type selection with the downstream retrieval tasks.

5.4 Ablation Study

We conduct an ablation study to evaluate the effects of the components of Prompt Module and Event Selection Module in LeClari (Table 6). First we find that LeClari performs worse significantly when the prompts exclude the event descriptions (just with event types), indicating that the descriptions provide useful information to help LLMs understand the event types. Then we analyze the variations which exclude one of the Transformers or one input embedding from Event Selection Module. First we find that when we exclude any one Transformer from Event Selection Module, LeClari experiences significantly a large performance degradation. It shows that Event Selection Module models the interaction information effectively between the event types and the conversations/candidate cases by these two Transformers, respectively. In addition, these three input embeddings also contribute to selecting better event types, especially the semantic embeddings, indicating that LeClari effectively models the relationships of legal event types, conversational context, and potential candidate cases.

Table 6: Ablation Study on LeCaRD. † denotes that LeClari performs significantly better than the variations at 0.05 level with a two-tailed pairwise t-test.

Method	ChatGPT+BERT-Crime			ChatGPT+LawFormer		
	MAP	P@5	NDCG@10	MAP	P@5	NDCG@10
LeClari	**0.4226**	**0.2860**	**0.6217**	**0.3475**	**0.2640**	**0.5972**
- Desc in Prompts	0.3912†	0.2340†	0.5711†	0.3016†	0.2310†	0.5787†
- Event-Conversation	0.3691†	0.2110†	0.5459†	0.2936†	0.2110†	0.5433†
- Event-Candidate	0.3846†	0.2290†	0.5671†	0.3134†	0.2290†	0.5599†
- semantic emb	0.3931†	0.2380†	0.5715†	0.3176†	0.2320†	0.5677†
- selected emb	0.4049†	0.2510†	0.5897†	0.3295†	0.2510†	0.5822†
- ranking emb	0.4081†	0.2520†	0.5875†	0.3325†	0.2490†	0.5841†

Table 7: A case study to compare the clarifying questions by "w/o Event" and LeClari. The three key points of query case for legal case retrieval and their corresponding answers are highlighted in boldface and distinguished by different colors.

<table>
<tr><td colspan="2"><b>Query Case</b>: On the evening of August 25th, 2013, <b>the defendant A discussed with co-defendant B about purchasing crystal meth</b>. On August 27th of the same year, <b>defendant A gave B RMB 28,000 to buy methamphetamine</b>. On August 28th, B bought 1 kilogram of methamphetamine from C in the same village for RMB 22,000 and gave it to A. From October 2015 until his arrest, <b>defendant A provided methamphetamine to D twice and to E three times</b>.</td></tr>
<tr><td colspan="2"><b>Initial Query</b>: Defendant A engaged in drug trafficking.</td></tr>
<tr><th>w/o Event+ChatGPT</th><th>LeClari+ChatGPT</th></tr>
<tr><td>q'_1: Can you tell us the specific amount of drugs that defendant A sold to others?
a'_1: <b>Defendant A provided methamphetamine to D and E for consumption</b>, but the specific quantity was not provided.</td><td>q'_1 (Event: "<b>Buying</b>"): Did defendant A purchase or produce the drugs, and what was the type and quantity of drugs involved?
a'_1: <b>Defendant A purchased the drugs</b>, which were methamphetamine, and the quantity was 1 kilogram.</td></tr>
<tr><td>q'_2: Did defendant A engage in drug trafficking before being arrested? If so, what was the specific quantity of drugs sold? (<b>Duplicated</b>)
a'_2: Defendant A provided drugs to D twice and to E three times before being arrested, indicating that he engaged in drug trafficking.</td><td>q'_2 (Event: "<b>Collusion</b>"): Did anyone else conspire with defendant A in drug trafficking?
a'_2: <b>Defendant A conspired with co-defendant B to purchase drugs</b>, and C in the same village was involved as the seller.</td></tr>
<tr><td>q'_3: Does defendant A have a criminal record? (<b>Useless</b>)
a'_3: It was not mentioned whether defendant A has a criminal record.</td><td>q'_3 (Event: "<b>Taking_drugs</b>"): Have defendant A's buyers already consumed the drugs?
a'_3: <b>Defendant A provided drugs to D twice and to E three times</b>, indicating that the buyers have already consumed the drugs.</td></tr>
</table>

Table 8: Comparison between MLE and RAML on LeCaRD. We highlight the better results in boldface and ‡ denotes the best results among them with different target metrics.

Target Metric	Training Strategy	ChatGPT+BERT-Crime			ChatGPT+LawFormer		
		MAP	P@5	NDCG@10	MAP	P@5	NDCG@10
MAP	MLE	0.4028	0.2520	0.5839	0.3235	0.2260	0.5626
	RAML	**0.4311**‡	**0.2820**	**0.6155**	**0.3524**‡	**0.2590**	**0.5942**
P@5	MLE	0.3878	0.2610	0.5863	0.3145	0.2410	0.5607
	RAML	**0.4185**	**0.2910**‡	**0.6176**	**0.3419**	**0.2730**‡	**0.5887**
NDCG@10	MLE	0.3925	0.2540	0.5918	0.3170	0.2350	0.5699
	RAML	**0.4226**	**0.2860**	**0.6217**‡	**0.3475**	**0.2640**	**0.5972**‡

5.5 Analysis on Training Strategies

Here we compare the two training strategies (i.e., MLE and RAML) when using different target ranking metrics on LeCaRD. It is obvious that RAML performs better than MLE. This is mainly because the MLE learning criterion brings the discrepancy between training and test, leading to overfitting on the ground-truth labels and reduced generalization ability. And RAML introduces the alternative outputs beyond the ground truth and overcomes this issue effectively. In addition, we find that when we utilize MAP as the target metric to generate the target distribution of event types, LeClari achieves higher MAP than when using the other two target metrics: P@5 and NDCG@10. And LeClari can also achieve the best P@5 and NDCG@10 by utilizing themselves as the target metrics, respectively. It demonstrates that we can use our most concerned metric as the target metric to derive the target distribution.

5.6 Case Study

We conduct a case study to show the clarifying questions generated by "w/o Event" and LeClari through ChatGPT (ref. Table 7). Here we highlight the three key points for legal case retrieval: A purchased the drugs rather than producing them by himself, A had a co-defendant B and A has sold the drugs. Here we find that when ChatGPT generates the clarifying questions directly without event type information, some of the questions ask about duplicated contents (e.g., q'_2) and some of them are useless for legal case retrieval (e.g., q'_3). This is consistent with the conclusion of our preliminary study. And LeClari incorporates three appropriate event types into the LLMs prompts and generates three clarifying questions corresponding to the three key points, respectively. It indicates that LeClari can select appropriate event types to cover the key points in the query case.

6 CONCLUSION

In this paper, we first conducted a preliminary study to show that generating clarifying questions in legal conversational search with SOTA LLMs (e.g., GPT-4) often suffers from several problems such as duplication and low-utility contents. Therefore, we leverage the legal event schema LEVEN to address these problems and propose a novel conversational search model LeClari with a Prompt Module and an Event Selection Module. The former defines a prompt with legal event for clarifying question generation and the latter selects potential event types by modeling the relationships of legal event types, conversational context, and potential candidate cases. We employ the RAML for the model learning to directly optimize the legal case retrieval metrics. Empirical results showed that our model can significantly outperform the state-of-the-art event selection methods. In the future work, we will make the model decide when to stop asking clarifying questions dynamically.

ACKNOWLEDGEMENTS

This work is supported by Quan Cheng Laboratory (Grant No. QCLZD202301) and the Natural Science Foundation of China (Grant No. 62002194).

REFERENCES

[1] Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval*. 475–484.

[2] Peter Auer. 2002. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research* 3, Nov (2002), 397–422.

[3] Nicholas J Belkin, Colleen Cool, Adelheit Stein, and Ulrich Thiel. 1995. Cases, scripts, and information-seeking strategies: On the design of interactive information retrieval systems. *Expert systems with applications* 9, 3 (1995), 379–395.

[4] Ben Carterette, Evangelos Kanoulas, Mark M. Hall, and Paul D. Clough. 2014. Overview of the TREC 2014 Session Track. In *Proceedings of The Twenty-Third Text REtrieval Conference, TREC 2014 (NIST Special Publication)*.

[5] Yanjiao Chen, Yuxuan Xiong, Bulou Liu, and Xiaoyan Yin. 2019. TranGAN: Generative adversarial network based transfer learning for social tie prediction. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 1–6.

[6] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards conversational recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 815–824.

[7] John Doyle. 1992. WESTLAW and the American Digest Classification Scheme. *Law Library Journal* 84 (1992), 229.

[8] Angel Sancho Ferrer, Carlos Fernández Hernández, and Pierre Boulat. 2014. LEGAL SEARCH: foundations, evolution and next challenges. The Wolters Kluwer experience. *Revista Democracia Digital e Governo Eletrônico* 1 (2014), 120–132.

[9] Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin* 76, 5 (1971), 378.

[10] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964* (2020).

[11] 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).

[12] Haitao Li, Weihang Su, Changyue Wang, Yueyue Wu, Qingyao Ai, and Yiqun Liu. 2023. THUIR@ COLIEE 2023: Incorporating Structural Knowledge into Pre-trained Language Models for Legal Case Retrieval. *arXiv preprint arXiv:2305.06812* (2023).

[13] Haitao Li, Changyue Wang, Weihang Su, Yueyue Wu, Qingyao Ai, and Yiqun Liu. 2023. THUIR@ COLIEE 2023: More Parameters and Legal Knowledge for Legal Case Entailment. *arXiv preprint arXiv:2305.06817* (2023).

[14] Bulou Liu, Bing Bai, Weibang Xie, Yiwen Guo, and Hao Chen. 2022. Task-optimized User Clustering based on Mobile App Usage for Cold-start Recommendations. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3347–3356.

[15] Bulou Liu, Yiran Hu, Yueyue Wu, Yiqun Liu, Fan Zhang, Chenliang Li, Min Zhang, Shaoping Ma, and Weixing Shen. 2023. Investigating Conversational Agent Action in Legal Case Retrieval. In *European Conference on Information Retrieval*. Springer, 622–635.

[16] Bulou Liu, Chenliang Li, Wei Zhou, Feng Ji, Yu Duan, and Haiqing Chen. 2020. An attention-based deep relevance model for few-shot document filtering. *ACM Transactions on Information Systems (TOIS)* 39, 1 (2020), 1–35.

[17] Bulou Liu, Yueyue Wu, Yiqun Liu, Fan Zhang, Yunqiu Shao, Chenliang Li, Min Zhang, and Shaoping Ma. 2021. Conversational vs Traditional: Comparing Search Behavior and Outcome in Legal Case Retrieval. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1622–1626.

[18] Bulou Liu, Yueyue Wu, Fan Zhang, Yiqun Liu, Zhihong Wang, Chenliang Li, Min Zhang, and Shaoping Ma. 2022. Query Generation and Buffer Mechanism: Towards a better conversational agent for legal case retrieval. *Information Processing & Management* 59, 5 (2022), 103051.

[19] Yixiao Ma, Yunqiu Shao, Bulou Liu, Yiqun Liu, Min Zhang, and Shaoping Ma. 2021. Retrieving Legal Cases from a Large-scale Candidate Corpus. In *Proceedings of the 18th International conference on Artificial Intelligence and Law*.

[20] 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.

[21] John O McGinnis and Russell G Pearce. 2019. The great disruption: How machine intelligence will transform the role of lawyers in the delivery of legal services. *Actual Problems of Economics and Law* 13 (2019), 1230.

[22] Mohammad Norouzi, Samy Bengio, Navdeep Jaitly, Mike Schuster, Yonghui Wu, Dale Schuurmans, et al. 2016. Reward augmented maximum likelihood for neural structured prediction. *Advances In Neural Information Processing Systems* 29 (2016).

[23] Robert Nowak. 2008. Generalized binary search. In *2008 46th Annual Allerton Conference on Communication, Control, and Computing*. IEEE, 568–574.

[24] OpenAI. 2022. Introducing ChatGPT. (2022).

[25] OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL]

[26] Filip Radlinski and Nick Craswell. 2017. A theoretical framework for conversational search. In *Proceedings of the 2017 conference on conference human information interaction and retrieval*. 117–126.

[27] Daniel E Rose and Richard K Belew. 1989. Legal information retrieval a hybrid approach. In *Proceedings of the 2nd international conference on Artificial intelligence and law*. 138–146.

[28] Manavalan Saravanan, Balaraman Ravindran, and Shivani Raman. 2009. Improving legal information retrieval using an ontological framework. *Artificial Intelligence and Law* 17, 2 (2009), 101–124.

[29] Yunqiu Shao, Bulou Liu, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. THUIR@ COLIEE-2020: Leveraging Semantic Understanding and Exact Matching for Legal Case Retrieval and Entailment. *arXiv preprint arXiv:2012.13102* (2020).

[30] 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.

[31] Yunqiu Shao, Yueyue Wu, Yiqun Liu, Jiaxin Mao, Min Zhang, and Shaoping Ma. 2021. Investigating User Behavior in Legal Case Retrieval. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 962–972.

[32] Paul Solomon. 1997. Conversation in information-seeking contexts: A test of an analytical framework. *Library & Information Science Research* 19, 3 (1997), 217–248.

[33] Howard Turtle. 1995. Text retrieval in the legal world. *Artificial Intelligence and Law* 3, 1 (1995), 5–54.

[34] Marc Van Opijnen and Cristiana Santos. 2017. On the concept of relevance in legal information retrieval. *Artificial Intelligence and Law* 25, 1 (2017), 65–87.

[35] Xiaozhi Wang, Xu Han, Zhiyuan Liu, Maosong Sun, and Peng Li. 2019. 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)*. 998–1008.

[36] Zhenduo Wang, Yuancheng Tu, Corby Rosset, Nick Craswell, Ming Wu, and Qingyao Ai. 2023. Zero-shot Clarifying Question Generation for Conversational Search. In *Proceedings of the ACM Web Conference 2023*. 3288–3298.

[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] 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*. 183–201.

[39] Feng Yao, Jingyuan Zhang, Yating Zhang, Xiaozhong Liu, Changlong Sun, Yun Liu, and Weixing Shen. 2023. Unsupervised Legal Evidence Retrieval via Contrastive Learning with Approximate Aggregated Positive. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 4783–4791.

[40] Hamed Zamani, Susan Dumais, Nick Craswell, Paul Bennett, and Gord Lueck. 2020. Generating clarifying questions for information retrieval. In *Proceedings of The Web Conference 2020*. 418–428.

[41] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W Bruce Croft. 2018. Towards conversational search and recommendation: System ask, user respond. In *Proceedings of the 27th acm international conference on information and knowledge management*. 177–186.

[42] Haoxi Zhong, Zhengyan Zhang, Zhiyuan Liu, and Maosong Sun. 2019. Open chinese language pre-trained model zoo. *Technical report* (2019).

[43] Jie Zou, Jimmy Huang, Zhaochun Ren, and Evangelos Kanoulas. 2022. Learning to Ask: Conversational Product Search via Representation Learning. *ACM Transactions on Information Systems* 41, 2 (2022), 1–27.

[44] Jie Zou and Evangelos Kanoulas. 2019. Learning to ask: Question-based sequential Bayesian product search. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 369–378.

[45] Jie Zou and Evangelos Kanoulas. 2020. Towards question-based high-recall information retrieval: Locating the last few relevant documents for technology-assisted reviews. *ACM Transactions on Information Systems (TOIS)* 38, 3 (2020), 1–35.