Pre-training for Legal Case Retrieval Based on Inter-Case Distinctions

WEIHANG SU, Quan Cheng Laboratory, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, China
QINGYAO AI*, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, China
YUEYUE WU, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, China
ANZHE XIE, School of Electronics Engineering and Computer Science, Peking University, China
CHANGYUE WANG, Quan Cheng Laboratory, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, China
YIXIAO MA, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, China
HAITAO LI, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, China
ZHIJING WU, School of Computer Science and Technology, Beijing Institute of Technology, China
YIQUN LIU, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, China
MIN ZHANG, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, China

Legal case retrieval aims to help legal workers find relevant cases related to their cases at hand, which is important for the guarantee of fairness and justice in legal judgments. While recent advances in neural retrieval methods have significantly improved the performance of open-domain retrieval tasks (e.g., Web search), their advantages haven't been observed in legal case retrieval due to their thirst for annotated data. As annotating

*Corresponding author

Authors' addresses: Weihang Su, swh22@mails.tsinghua.edu.cn, Quan Cheng Laboratory, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, Beijing, China; Qingyao Ai, aiqy@tsinghua.edu.cn, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, Beijing, China; Yueyue Wu, wuyueyue1600@gmail.com, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, Beijing, China; Anzhe Xie, xaz@stu.pku.edu.cn, School of Electronics Engineering and Computer Science, Peking University, Beijing, China; Changyue Wang, changyue20@mails.tsinghua.edu.cn, Quan Cheng Laboratory, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, Beijing, China; Yixiao Ma, ma-yx16@tsinghua.org.cn, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, Beijing, China; Haitao Li, liht22@mails.tsinghua.edu.cn, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, Beijing, China; Zhijing Wu, zhijingwu@bit.edu.cn, School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China; Yiqun Liu, yiqunliu@tsinghua.edu.cn, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, Beijing, China; Min Zhang, z-m@tsinghua.edu.cn, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, Beijing, China.

XXXX-XXXX/2024/9-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

large-scale training data in legal domains is prohibitive due to the need for domain expertise, traditional search techniques based on lexical matching such as TF-IDF, BM25, and Query Likelihood are still prevalent in legal case retrieval systems. While previous studies have designed several pre-training methods for IR models in open-domain tasks, these methods are usually suboptimal in legal case retrieval because they cannot understand and capture the key knowledge and data structures in the legal corpus. To this end, we propose a novel pre-training framework named Caseformer that enables the pre-trained models to learn legal knowledge and domain-specific relevance-matching patterns in legal case retrieval without any human-labeled data. This framework is designed to support both dense retrieval models and neural re-ranking models. Through three unsupervised learning tasks, Caseformer is able to capture the special language, document structure, and relevance-matching patterns of legal case documents, making it a strong backbone for downstream legal case retrieval tasks. Experimental results show that our model has achieved state-of-the-art performance in both zero-shot and fine-tuning settings. Also, experiments on both Chinese and English legal datasets demonstrate that the effectiveness of Caseformer is language-independent in legal case retrieval.

CCS Concepts: • **Information systems** → **Similarity measures**; **Retrieval models and ranking**.

Additional Key Words and Phrases: Legal Case Retrieval, Pre-training Methods, Contrastive Learning

ACM Reference Format:
Weihang Su, Qingyao Ai, Yueyue Wu, Anzhe Xie, Changyue Wang, Yixiao Ma, Haitao Li, Zhijing Wu, Yiqun Liu, and Min Zhang. 2024. Pre-training for Legal Case Retrieval Based on Inter-Case Distinctions. 1, 1 (September 2024), 28 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Legal case retrieval helps legal workers find relevant cases related to their cases at hand, which is important for the fairness and justice of legal judgments. As of today, most legal case retrieval systems still rely on simple bag-of-words retrieval models to retrieve documents based on users' queries [44]. While recent advances in neural retrieval methods have significantly improved the performance of open-domain retrieval tasks (e.g., Web search) [24, 39], their advantages haven't been observed in legal case retrieval due to their thirst for annotated data. The training of state-of-the-art neural retrieval models usually requires millions or even billions of annotated query-document pairs to achieve desired effectiveness and reliability. In the legal domain, however, creating such large-scale training data is prohibitive due to the need for domain experts as assessors. For instance, in countries that adopt the civil law system [1] (e.g., Germany, Japan, and China), prior cases are not required to be involved in judgment, which means that there are no similar cases recorded in judgment documents, extra efforts are needed to create annotated datasets for the training of neural retrieval models. In China, the largest case retrieval dataset LeCaRD [36] only contains 107 labeled queries, which is far from enough to train an effective neural retriever. Therefore, statistical ranking models based on lexical matching such as TF-IDF [42], BM25 [43], and Query Likelihood [66] are still the mainstream techniques adopted by legal case retrieval systems [35, 44].

To address the lack of supervision data in open-domain retrieval tasks such as Web search, pre-training methods that initialize neural retrieval models with unsupervised training signals have attracted much attention. As the pre-training and fine-tuning paradigms have achieved state-of-the-art performance in NLP tasks [14, 31, 57, 63, 64], the IR community begins to explore pre-training methods tailored for IR [6, 8, 15, 33, 34, 38, 47, 51]. These PLMs are usually pre-trained on general domain corpus such as Wikipedia and have achieved better performance compared with their original versions such as BERT [14] and RoBERTa [31] in most retrieval and re-ranking tasks.

Unfortunately, existing PLMs tailored for general IR tasks do not fit the needs of legal case retrieval due to their incapability of legal document understanding. The definition of information

[1] Civil law is the most widely adopted legal system in the world. It refers to structuring legal systems around broad codes and detailed statutes that determine individuals' rights and obligations.

relevance in the legal field is different from general ad-hoc retrieval tasks [36, 45]. Besides the text similarity information used by open-domain retrieval models, case relevance in legal retrieval cares more about the legal similarity and relationships between legal elements [45]. While some researchers have explored the possibility of adapting existing Pre-trained Language Models (PLMs) for legal data [5, 26, 59], a significant research gap exists in the modeling of legal relevance across different legal case documents. This aspect is particularly important for legal case retrieval because legal documents are interconnected by charges and cases with similar properties. Given this, the current approaches, though advanced, fall short of fully harnessing the potential of PLMs for legal case retrieval tasks, highlighting a substantial opportunity for enhanced methodological development and improved performance in this specialized field.

To this end, we propose a novel pre-train framework named Caseformer that enables the pre-trained models to learn legal knowledge and relevance-matching patterns between legal cases from raw legal corpora without any human-labeled data. This pre-training framework is designed to support both dense retrieval models and neural re-ranking models. Specifically, we propose three pre-training tasks: 1) Legal LAnguage Modeling (LAM), 2) Legal Judgment Prediction (LJP), and 3) Factual Description Matching (FDM). In the LAM task, we train the model to internalize the distinctive linguistic patterns and characteristics of the legal domain. In the LJP task, we train the model to measure relevance and connections between cases based on their similarity in legal judgments. Then, in the FDM task, we further train the model to measure case relevance based on the similarity between the fact descriptions in different case documents. Through these three pre-training tasks, Caseformer can capture domain-specific linguistic patterns, structures, and relevance-matching patterns across legal case documents, making it a strong backbone for downstream legal case retrieval tasks. Experimental results on three legal case retrieval datasets (both in English and Chinese) and two legal case relevance judgment datasets (both in English and Chinese) show that the re-ranking and retrieval models based on Caseformer can achieve state-of-the-art performance in zero-shot and fine-tuning settings.

To summarize, the contributions of this paper are as follows:

- We propose a novel pre-training framework, Caseformer[2], to solve the data-hungry problem of existing neural retrieval and re-ranking models in legal case retrieval scenarios.
- We propose three pre-training objectives that enable the proposed models to capture legal case documents' special language features, structure information, and relevance patterns between legal case documents.
- We evaluate the performance of our framework on multiple legal case retrieval datasets, and the results show that Caseformer outperforms baselines in various settings.

2 RELATED WORK

2.1 Pre-training Methods for IR

As pre-trained language models have achieved great success in the NLP field, the IR community begins to utilize PLMs to solve downstream IR tasks [7, 9, 16, 27–30, 37, 65], and design pre-training methods tailored for information retrieval [6, 8, 15, 22, 26, 33, 34, 38, 49, 51]. The key idea of existing IR pre-training methods is to construct pseudo-relevant query-document pairs from unlabeled corpora. For example, Chang et al. [6] designed three pre-training tasks based on Wikipedia: Inverse Cloze Task (ICT), Body First Selection (BFS), and Wiki Link Prediction (WLP). In these tasks, a sentence from a passage is randomly selected as a query, and the selected passage is defined as the corresponding relevant document. Also using Wikipedia as the pre-training corpus, Ma et al. [38] utilizes the hyperlinks and their corresponding anchor text to train a re-ranking model

[2]We open source the entire project code in this link: https://github.com/oneal2000/Caseformer

named HARP. Webformer [22] utilizes the structural information of Wikipedia web pages and designs four pre-training tasks. Instead of using Wikipedia as the training corpus, PROP [33] and B-PROP [34] are pre-trained on the plain text by the Representative Words Prediction (ROP) task. They assume that a sampled word set from a document with a higher query likelihood score is more "representative" of that document. Based on this assumption, they train the model to predict pairwise preference between two sampled word sets and achieve state-of-the-art performance.

In summary, the pre-training objectives of current Information Retrieval Pre-trained Language Models (IR PLMs) are mostly designed for open domain tasks without special focus on any types of documents or domains. However, as shown in this paper, pre-training retrieval models without considering the special structures and characteristics of legal documents often lead to suboptimal performance in legal case retrieval. This motivates us to study how to construct and incorporate legal domain knowledge into the pre-training of legal retrieval models.

2.2 Legal Domain Pre-training

As PLMs pre-trained in generic domains don't work well on legal tasks, several studies have explored the possibility of constructing legal-specific pre-training models for legal tasks. Xiao et al. [59] proposed a legal domain pre-trained language model named Lawformer which is initialized by Longformer [2] as the basic encoder. Lawformer is pre-trained on millions of case documents published by China Judgments Online [3] and has good performance after fine-tuning on downstream tasks. However, Lawformer is essentially a re-training of existing PLMs on the legal documents, which limits its capability in modeling domain-specific data structures.

Chalkidis et al. [5] propose a legal domain pre-trained model named Legal-BERT. They explore three strategies for using BERT to solve legal tasks: 1) use the original BERT directly, 2) adapt the original BERT by additional training on the legal domain, and 3) pre-train BERT from scratch on legal corpora. They found that further training in the legal domain is better than using the original BERT directly. Nonetheless, Legal-BERT directly uses the official BERT code in the pre-training stage and no changes have been made to adapt the unique characteristics of the legal field.

The most related study to this paper is the legal PLM proposed by Li et al. [26] named SAILER. SAILER is designed to model a single legal case document through the encoder-decoder architecture. This architecture is adept at modeling and capturing the dependency between the Fact Description[4] section and other sections within a legal case document, which allows SAILER to leverage the logical relationships in the structures of a single legal document. However, it's crucial to note that SAILER's focus is predominantly on modeling the representation of individual legal case documents. It does not extend to examining the interrelationships between different legal documents. This limitation presents a gap in the current approach, as understanding the relevance among different legal documents is essential for comprehensive legal analysis and case retrieval. Our work aims to bridge this gap by proposing a method that not only accounts for the intricate details within individual case documents but also explores and models the relationships between different legal documents.

In summary, while existing work has successfully applied PLMs to the legal domain, they notably fall short in capturing the legal-level relevance among various legal case documents. This oversight underscores a significant opportunity for advanced methodological development and enhanced performance in the legal field. Our research is strategically positioned to bridge this gap. By employing three unsupervised tasks, we aim to enable PLMs to effectively model the relevance between legal documents.

[3] https://wenshu.court.gov.cn

[4] The Fact Description section describes the facts and circumstances of a legal case.

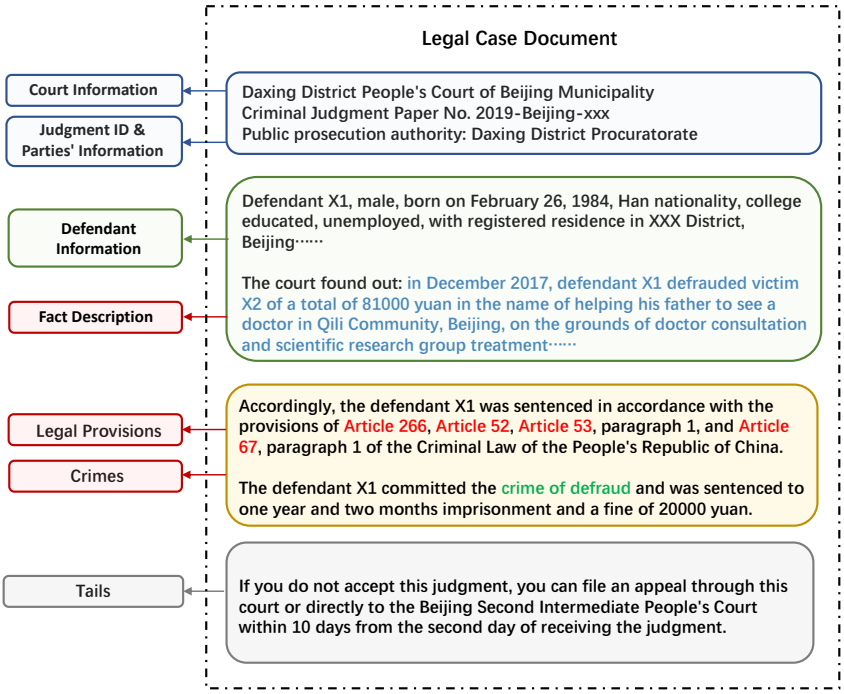

Fig. 1. An example of the writing organization of legal documents and their semi-structured information. The LJP and FDM tasks utilize three types of information: Fact Description, Legal Provisions, and Crimes which are highlighted in blue, red, and green fonts respectively.

3 PROBLEM FORMULATION

The legal case retrieval task aims to retrieve relevant cases (represented by case documents) given the fact description of an unjudged query case. More specifically, given a query case q and a set of candidate cases $C = \{c_1, c_2, ..., c_n\}$, where $n \in N^+$, let r_i be the Bernoulli variable indicating whether c_i is relevant to q, then the task of legal case retrieval is to retrieve a set of cases $S = \{c_j | r_j = 1\}$.

As shown in figure 1, a candidate legal case document usually consists of the following parts:

- **Court information** which provides detailed information about the court which produces the document. It typically includes the name of the court, the case number, the presiding judge's name, and the date of judgment.
- **Defendant information** which provides information about the individual or entity against whom the legal action is being taken. It usually includes the defendant's full name, gender, date of birth, nationality, and other relevant identifying details.
- **Fact Description** which describes the facts and circumstances of the case. It includes a comprehensive account of events, actions, or situations that led to the legal dispute. The fact description part of a well-written case document is usually clear, concise, and objective.
- **Legal Provisions** which describes the relevant laws, statutes, regulations, or legal provisions that apply to the case. It may include references to specific sections or articles of the law that are relevant to the issues at hand.
- **Crimes** which describes the specific crimes or offenses that the defendant is accused of committing which serves to identify the charges against the defendant.

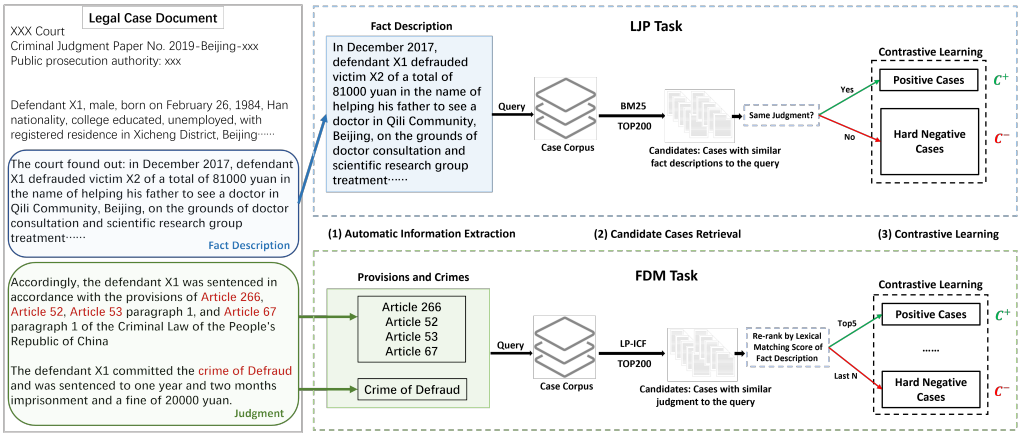

Fig. 2. Illustration of the proposed pre-training tasks of Caseformer. Generally, there are three main stages: (1) Automatic Information Extraction, (2) Candidate Cases Retrieval, (3) Contrastive Learning.

In legal case retrieval practice, the query (q) and candidate documents typically comprise only the Fact Description part. Our work adopts this configuration, assuming that queries and candidate documents are fact descriptions extracted from legal case documents. In most cases, the length of the fact description section in a legal case is smaller than the maximum input length of the pre-trained model. For the rare cases where the fact description exceeds the maximum input length, we utilize a truncated approach to handle it (details can be found in Section 5.5).

4 METHODOLOGY

In this section, we analyze the abilities of an ideal legal case retrieval model and discuss how we propose different pre-training tasks accordingly. To be specific, we introduce the model architecture in §4.1 and the details of training process in § 4.2, §4.3, and §4.4. The training process includes three pre-training tasks: Legal LAnguage Modeling (LAM), Legal Judgment Prediction (LJP), and Factual Description Matching (FDM). And then in section §4.5, we introduce the overall training objective of our framework.

4.1 Model Architecture

In practical retrieval systems, a two-stage pipeline is usually adopted to balance the overall effectiveness and efficiency. The first stage is the retrieval stage, where the relevant documents are recalled from an extensive corpus. This is followed by the re-ranking stage, where the documents recalled in the first stage are re-ranked according to their relevance to the query. The initial retrieval stage aims to swiftly find potentially relevant documents within the entire corpus. The re-ranking stage, although more time-consuming, enables a more precise evaluation of each document's relevance to the query.

Existing Pre-trained Language Model (PLM) based search methods can be typically categorized into two architectures: dual-encoder and cross-encoder. Dual-encoders encode the query and candidate documents separately without considering the token-level interactions. As the corpus can be pre-encoded into dense representation vectors, dual-encoders are widely used in the first-stage retrieval. Cross-encoders, in contrast, concatenate queries and documents as a single input for the PLM, enabling detailed token-level interaction, thus enhancing the ranking accuracy. However,

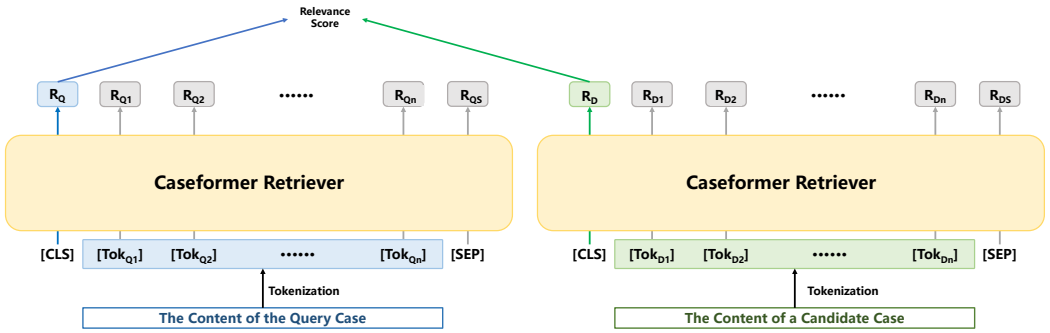

Fig. 3. Illustration of the Caseformer-Retriever model's architecture. The process begins with word piece tokenization, appending special tokens [CLS] at the start and [SEP] at the end. Then the query case and the candidate case are encoded into dense vector representations. Subsequently, the relevance score is calculated based on these representations.

due to significant inference latency, cross-encoders are confined to re-ranking tasks within smaller datasets.

Applying this two-stage retrieval approach is also beneficial for legal case retrieval. The first stage ensures that the system can quickly process requests by filtering through a large legal case corpus. The second stage, while more time-consuming, enhances the quality of the results by carefully evaluating the relevance of each case. Therefore, we introduce both dual-encoder and cross-encoder architectures: Caseformer-Retriever and Caseformer-Re-ranker, designed for each stage respectively which are shown in figure 3 and figure 4. This section will introduce the architectures of these two models.

4.1.1 Caseformer Retriever. We use the Transformer-encoder architecture (structurally the same as BERT [14]) for the implementation of Caseformer-Retriever. Following the BERT's tokenization methodology [14], we utilize word piece tokenization to convert the input text into discrete tokens. This tokenization process involves appending a [CLS] token at the start and a [SEP] token at the end of the token sequence. These tokens are then inputted into the Transformer-encoder [57] to generate a contextualized embedding vector for each token. Following the setting of DPR [24], the embedding corresponding to the [CLS] token serves as the comprehensive representation of the legal case. This encoding process of a legal case C can be formulated as follows:

$$Input\_ids = [CLS]\ tokenizer(C)\ [SEP] \tag{1}$$

$$Rep(C) = transformer_{[CLS]}(Input\_ids) \tag{2}$$

where $tokenizer(x)$ utilizes word piece tokenization to convert the input text x into discrete tokens, $transformer_{[CLS]}(\cdot)$ first encode the input with a transformer model and then extract the embedding vector of the [CLS] as the final representation of the input data. After acquiring the representations, we regard the inner product of two cases (C_i and C_j) as their relevance score:

$$s(C_i, C_j) = Rep(C_i)^\top \cdot Rep(C_j) \tag{3}$$

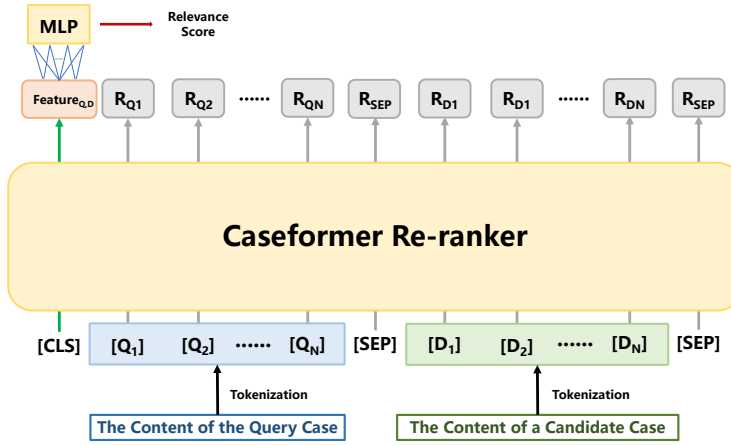

Fig. 4. Illustraion of the Caseformer-reranker architecture. The process begins with word piece tokenization, appending special tokens [CLS] at the start, [SEP] for separation and at the end. The [CLS] token's embedding represents the interaction feature between the query and candidate case. A multi-layer perceptron (MLP) layer then maps this feature vector to a relevance score.

In summary, the primary objective of the Caseformer-Retriever model is to model each legal case into a dense vector representation. The training process for the Caseformer-Retriever focuses on refining these legal case representations to optimize the retrieval effectiveness, which is detailed in § 4.2, §4.3, and §4.4.

4.1.2 Caseformer Re-ranker. We use the Transformer-encoder architecture for the implementation of Caseformer-reranker. Following the BERT's tokenization methodology [14], we utilize word piece tokenization. This tokenization process begins with adding a [CLS] token at the beginning, a [SEP] token to seperate the tokenized query and candidate case, and a [SEP] token at the end of the token sequence. These tokens are then inputted into the Transformer-encoder [57] to generate a contextualized embedding vector for each token. Following the setting of [13, 20], the embedding corresponding to the [CLS] token serves as the representation of the token-level interaction feature between the query and the candidate case. Subsequently, a multi-layer perceptron (MLP) layer maps the relevance feature vector to its corresponding relevance score. This process can be formulated as follows:

$$X_{C_Q,C_D} = [CLS]tokenizer(C_Q)[SEP]tokenizer(C_D)[SEP] \tag{4}$$

$$f(C_Q, C_D) = transformer_{[CLS]}(X_{Q,D}) \tag{5}$$

$$s(C_Q, C_D) = MLP(f(C_Q, C_D))) \tag{6}$$

where C_Q is the query case and C_D is the candidate case, $tokenizer(x)$ utilizes word piece tokenization to convert the input text x into discrete tokens, $MLP(\cdot)$ is a multi-layer perceptron that projects the relevance feature vector f to a relevance score s. The training process for the Caseformer-Reranker is detailed in § 4.2, §4.3, and §4.4.

4.2 Legal LAnguage Modeling (LAM) Task: Understanding Legal Language

As discussed previously, an ideal legal case retrieval model should have the ability to capture and understand the domain language used in legal documents. As illustrated in Figure 1, legal documents often contain specialized terminology, expressions, and content structures that are rarely observed in general domain documents. We consider these specific professional terms, expressions, and writing structures within the legal field as legal language.

To address the limitations of existing PLMs in legal case retrieval, we propose a pre-training task named Legal LAnguage Modeling (LAM). Specifically, we first pre-train the model on official law books (e.g., official criminal code, official judicial interpretation, etc.) with the Mask Language Modeling (MLM) task. For each document in the corpus, we tokenize the document and then divide the tokenized document into input sequences $X = [x_1, x_2, x_3, \ldots, x_n]$, where n is the maximum input length of the model. We then use a dynamic masking strategy [31] to randomly replace the tokens in X with the special token [MASK]. The MLM loss L can be calculated as:

$$\mathcal{L}_{MLM} = -\sum_{\hat{x} \in m(X)} \log(P(\hat{x}|X_{\backslash m(X)})) \tag{7}$$

where X denotes the input sequence, $m(X)$ and $X_{\backslash m(X)}$ are the masked word set and the rest words in X, respectively, $\log(P(\hat{x}|X_{\backslash m(X)}))$ is the model predicted probability of token $\hat{x}$. The model is trained to minimize this MLM loss by updating its parameters through backpropagation and gradient descent optimization algorithms. The official law books we used here are issued by the government of a country with strict language organization and reliable content. They usually contain a comprehensive range of professional terms and expressions utilized in the legal domain. Therefore, through conducting MLM training on this corpus, we want the model to acquire an accurate understanding of the meaning associated with professional terms and expressions used in the legal field.

To enhance the model's acquisition of legal knowledge and its ability to adapt to the distinctive writing structure within the legal domain, we additionally conduct pre-training on legal case documents with the MLM task. These legal documents encompass a wealth of legal knowledge, such as the exposition of fundamental facts, legal provisions, and criminal offenses. By training the model on these documents, we want to adapt it to the specific writing structure characteristic of the legal field, while also acquiring a comprehensive understanding of legal knowledge.

Overall, the idea of the LAM task is to pre-train retrieval models with MLM tasks on legal documents that provide comprehensive definitions and explanations of legal terminology and concepts. With LAM, our goal is to enhance the model's ability to understand legal language in fine grains so that it can better serve downstream training and applications.

4.3 Legal Judgment Prediction (LJP) Task: Measuring Judgment Similarity

Retrieval models pre-trained on general domain data are usually good at matching documents according to their semantic similarity in text. However, in the context of legal case retrieval, the models should not only consider semantic similarity but also evaluate the legal-level similarity between cases. This distinction arises from the unique needs of the legal field, where the assessment of case relevance consider both semantic and legal aspects. For example, consider the fact descriptions of the following cases,

Case 1:
On March 10, 2022, on X Street in K City, ***two*** *people with baseball hats gathered to fight after drinking.*

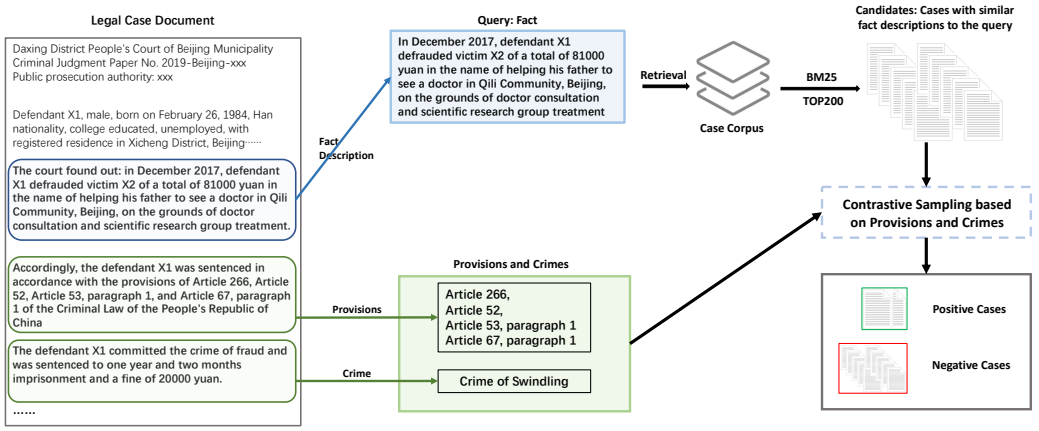

Fig. 5. The contrastive sampling strategy of the LJP task.

The police quickly arrived at the scene, controlled the people involved in the fight, and avoided further escalation of the situation. This case finally resulted in the ***reconciliation*** *of those two people.*

Case 2:
On March 10, 2022, on X Street in K City, ***twenty*** *people with iron baseball bats gathered to fight after drinking. The police quickly arrived at the scene, controlled the people involved in the fight, and avoided further escalation of the situation. This case finally resulted in* ***two deaths****.*

In the first case, the situation was quickly resolved, resulting in reconciliation between the parties involved. This suggests that it may not be a criminal case and might have been handled through alternative means such as mediation or civil proceedings. On the other hand, the second case, while having a similar fact description in terms of time, location, and gathering of individuals, it finally escalated into a major criminal case with two deaths involved. It can be seen that some cases with similar semantics could be fundamentally different at the legal level.

To teach the model to understand case relevance from legal perspectives, we train the model on legal case documents by proposing an unsupervised contrastive learning task named Legal Judgment Prediction (LJP). As shown in Figure 1, a standard case document consists of several parts including factual descriptions, legal provisions, and the crimes of the case which can be automatically extracted based on the writing structure of each case. The proposed Legal Judgment Prediction (LJP) task is illustrated in Figure 2. The basic assumption of LJP is that, **among cases with similar fact descriptions, cases with the same judgments are usually relevant to each other**. Specifically, we train the model to select cases with the same crimes and provisions from a series of cases with similar factual descriptions. Firstly, given a corpus consisting of legal case documents, we extract the factual description, committed crimes, and legal provisions of each case. Then a case Q is randomly selected from a case collection corpus and its factual description is used as the query. Based on the query Q, the BM25 method is adopted to compute the similarity between the fact descriptions of the query and the candidates, and recall the top 200 similar cases in terms of fact description. The recalled cases set is defined as C. **For each case c_i in C, if the crimes and legal provisions of c_i are the same as the query case Q, then we treat c_i as a positive**

example and add it to the set C^+, if not, c_i is defined as a negative example and added to the set C^-.

After sampling the positive and negative examples, we use a contrastive learning strategy to train both the re-ranking model and the retrieval model. **For the re-ranking model**, we use the cross-encoder architecture [39] to compute the relevance score between two legal case documents c_i and c_j:

$$X_{ij} = [CLS]c_i^f[SEP]c_j^f[SEP] \tag{8}$$

$$s(c_i, c_j) = MLP(transformer_{[CLS]}(X_{ij})) \tag{9}$$

where c_i^f is the factual description extracted from the c_i, $transformer_{[CLS]}(\cdot)$ first encode the input with a transformer model and then output the embedding vector of the [CLS] as the final representation of the input data. $MLP(\cdot)$ is a multi-layer perceptron that projects the [CLS] embedding to a relevance score s.

For the retrieval model, we use the dual-encoder architecture [24] to compute the dot product between two embedding vectors as the relevance score:

$$X(c) = [CLS]c^f[SEP] \tag{10}$$

$$Emb(X) = transformer_{[CLS]}(X) \tag{11}$$

$$s(c_i, c_j) = Emb(X(c_i))^\top \cdot Emb(X(c_j)) \tag{12}$$

where c^f is the factual description extracted from the input case c, $transformer_{[CLS]}(\cdot)$ outputs a contextualized vector for each token and we select the "[CLS]" vector as the embedding vector of a case. In Equation 12, we regard the inner products of case embeddings as the relevance score s.

For the loss function, we use the Softmax Cross Entropy Loss [1, 4, 20] to optimize the re-ranking and retrieval model, which is defined as:

$$\begin{aligned} &\mathcal{L}_{LJP}(Q, c^+, N) \\ &= -\log \frac{exp(s(Q, c^+))}{exp(s(Q, c^+) + \sum_{c^- \in N} exp(s(Q, c^-))} \end{aligned} \tag{13}$$

where s is the relevance score function which is defined in Equation 9 and Equation 12 for re-ranking and retrieval models respectively. Q is the query case, c^+ is a selected positive case and N is the set of selected negative cases.

Note that the matching of crimes and legal provisions may be influenced by typographical errors and non-standard writings, but in our dataset, such occurrences account for less than 1%. We have also employed a range of methods to address this issue, and the specific implementation details can be found in our open-source code[5]. This problem is primarily an engineering problem and is not the focus of this paper.

4.4 Factual Description Matching (FDM) Task: Measuring Factual Similarity

With the LJP task, we train the model to better distinguish cases based on their judgment information. On the other hand, an ideal case retrieval model should also be able to distinguish relevant and irrelevant case documents based on factual description information of legal cases. Therefore,

[5]https://github.com/caseformer/caseformer

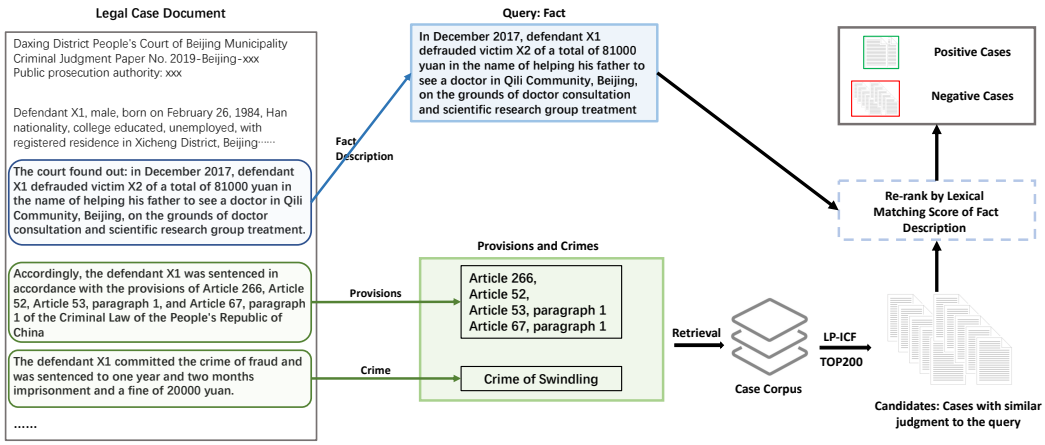

Fig. 6. The contrastive sampling strategy of the FDM task.

we propose the Factual Description Matching (FDM) task which is illustrated in Figure 2. Our assumption is that **in cases with similar judgments, cases with similar fact descriptions should be more relevant to each other than those that don't.** To be specific, we train the model to select relevant cases based on the fact description from a set of candidate cases with similar legal judgments. First, we propose a retrieval method named Legal Provision-Inverse Case Frequency (LP-ICF) to find cases with similar judgments. Specifically, given a legal case document, we extract its crimes (charges) and legal provisions from the judgment. Based on the crimes and legal provisions, the Legal Provision-Inverse Case Frequency (LP-ICF) method is defined as follows. Given a legal case document collection D, a case as the query (c_i), and a candidate case (c_j):

$$LP-ICF(c_i, c_j) = crime_{i,j} * \sum_{p \in P} log \frac{|D|}{freq(p, D)} \tag{14}$$

where P is the set of overlapping provisions between c_i and c_j, $|D|$ is the size of the collection D, $freq(p, D)$ is the number of the appearance of provision p in the collection D, and $crime_{i,j}$ is set to 0 if there's no overlap in the crimes between $case_i$ and $case_j$ and otherwise, is set to 1. The LP-ICF method can be understood as follows: first, calculate the inverse case frequency (ICF) of each legal provision based on its appearance. Then, according to the overlap of the two cases in crimes and legal provisions, their similarity score is calculated by Equation 14. In short, LP-ICF can recall a series of cases that are similar in judgment within a short time.

Based on the LP-ICF method, given a case Q, we first recall the top 200 relevant cases that are similar in judgment by LP-ICF. The recalled cases set is defined as C. **For each case c_i in C, we calculate the lexical relevance between the factual description of c_i and Q by BM25. Then randomly select a case from the top 5 of the BM25 ranking list as the positive example c^+ and select the last λ cases of the BM25 ranking list (where λ is an adjustable hyperparameter) and add them to the negative examples set C^-.**

After sampling the positive and negative examples, we use a contrastive learning strategy to train both the re-ranking model and the retrieval model. For the loss function, we still use the

Softmax Cross Entropy Loss to optimize our model, which is defined as:

$$\begin{aligned}&\mathcal{L}_{FDM}(Q, c^+, N)\\&= -\log \frac{exp(s(Q, c^+))}{exp(s(Q, c^+) + \sum_{c^- \in N} exp(s(Q, c^-))}\end{aligned} \tag{15}$$

where s is the relevance score function which is defined in Equation 9 and Equation 12 for re-ranking and retrieval models. Q is the query cases, c^+ is the selected positive case and N is the set of selected negative cases.

4.5 Final Training Objective

We combined the above three tasks as the final training objective, which is defined as follows:

$$\begin{aligned}&\mathcal{L}_{final}(Q, P_{LJP}, N_{LJP}, P_{FDM}, N_{FDM})\\&= \mathcal{L}_{LAM}(Q) + \mathcal{L}_{LJP}(Q, P_{LJP}, N_{LJP}) + \mathcal{L}_{FDM}(Q, P_{FDM}, N_{FDM})\end{aligned} \tag{16}$$

where Q is a case from the corpus, P_{LJP} is the positive example generated from the LJP task, N_{LJP} is the set of negative examples generated from the LJP task, and P_{FDM}, N_{FDM} are the positive example and negative examples generated from the FDM task. We conducted ablation experiments on all combinations of loss in Section 6.4.

5 EXPERIMENT SETUP

5.1 Pre-training Corpus

We select two types of legal corpus, one in Chinese and one in English, as the pre-training corpora for our experiments. For the Chinese version of Caseformer under the Chinese criminal law system, we first pre-train the model on the official law books[6] in the LAM task. Then in the LJP and FDM tasks, we pre-train the model on 5M case documents released by the Supreme Court of China. Based on this corpus, we generate around 800M pseudo-query-case pairs. For the English version of Caseformer, we first pre-train the model on the Indian Penal Code in the LAM task. Then in the LJP and FDM tasks, we pre-train the model on the ILSI [40] dataset which contains 66,090 legal cases from several major Indian Courts. The fact description and legal provisions of each case are provided in ILSI. Based on this corpus, we generate around 13M pseudo-query-case pairs. Note that rigorous checks confirm the complete absence of overlap between the pre-training and testing datasets, thus upholding the validity of our model's evaluation metrics.

5.2 Extraction of Facts, Crimes, Provisions and Crimes

Legal documents in mainland China adhere to a highly standardized format due to strict regulatory requirements. Each legal judgment document consistently contains three sequential sections: **Facts**, **Reasoning**, and **Decision**. These sections are delineated by specific phrases, enabling reliable automated extraction. For example:

- The **Facts** section typically begins with phrases like "After trial, it was found that…".
- The **Reasoning** section starts with phrases like "The court holds that…", with the content between the Facts and Reasoning sections constituting the factual description.
- The **Decision** section follows a standardized format such as "According to Article [Article Number] of [Law Name], the judgment is as follows…", detailing the crimes committed and the corresponding penalties.

[6]All corpus are uploaded to our anonymous Github link: https://github.com/caseformer/caseformer

Leveraging this standardization, we designed regular expression patterns to match the markers at the beginning of each document section. Through comprehensive analysis, we developed patterns that cover almost all standard cases. Documents that did not adhere to the standard formatting were filtered out during preprocessing. After excluding these non-standard documents, our automatic extraction process achieved an error rate of less than 1%. This approach ensured that the remaining documents could be reliably processed using lexical matching methods for the different sections.

5.3 Legal case retrieval Datasets

We evaluate the performance of Caseformer on the following datasets.

- **LeCaRD** [36] is the largest Chinese case retrieval dataset, consisting of 107 query cases and over 43000 candidate cases[7]. All the cases are adopted from criminal cases published by the Supreme People's Court of China. The queries and candidate documents in the LeCaRD dataset are the factual description part (introduced in Section 3) of a legal case.
- **CAIL-SCM** [60] is a case relevance judgment dataset provided by CAIL 2019. All the cases are published by China Judgments Online[8], an official website of the Chinese Legal System. Each data is composed of one query case and two candidate cases. For each legal case document, the title and fact description is provided. Both the queries and candidate documents in the CAIL-SCM dataset are the factual descriptions part of a legal case.
- **CAIL-LCR** [9] is a case retrieval dataset provided by CAIL 2022 consisting of 130 query cases and 100 candidate cases for each query case. The queries and candidate documents in the LeCaRD dataset are the factual description part of a legal case.
- **COLIEE 2020 Task1** [41] is an English version case retrieval task provided by COLIEE[10]. The training set contains 520 query cases and 200 candidate cases for each query case. The test set contains 130 query cases and 200 candidate cases for each query case. Both the queries and candidate documents in the COLIEE 2020 dataset are complete legal case documents with all parts listed in Figure 1.

5.4 Baselines

We consider four types of baselines for comparison, including traditional IR methods, pre-trained Language models on general domain data, PLMs tailored for IR, and pre-trained language models built with legal documents.

- **Traditional IR Methods**
  - **QL** [66] is a language model based on Dirichlet smoothing and has good performance on retrieval tasks.
  - **BM25** [43] is a highly effective retrieval model based on lexical matching that achieves good performance in retrieval tasks.

  For the implementation, we use the pyserini toolkit[11]. For the hyperparameter of BM25, we set $k1 = 3.8$ and $b = 0.87$[12]. Note that in our experiments, we use the scores of the BM25 and QL models to re-rank the candidate documents, rather than re-ranking the whole corpus.
- **Pre-trained Models tailored for IR**

[7]Note that LeCaRD provides two types of re-ranking tasks: the number of candidate documents is 30 and 100 respectively. The original paper [36] and Lawformer [59] use the setting of 30. We choose the setting of 100 (following the setting of SAILER [26]) which is more likely to distinguish the differences in models' performance.
[8]https://wenshu.court.gov.cn/
[9]https://github.com/china-ai-law-challenge/CAIL2022/tree/main/lajs
[10]https://sites.ualberta.ca/ rabelo/COLIEE2020/
[11]https://github.com/castorini/pyserini
[12]This is the best hyperparameter we got after parameter searching.

  - **PROP** [33] is a pre-trained model with cross-encoder architecture tailored for IR re-ranking tasks. It adopts the Representative Words Prediction (ROP) task to predict the pairwise preference between word sets. [13]
  - **SEED** [32] is a pre-trained text encoder for dense retrieval that achieves state-of-the-art performance.
  - **Condenser** [18]. Condenser [18] is a state-of-the-art pre-training architecture for dense retrieval. It leverages skip connections to consolidate textual information into dense vectors.
  - **coCondenser** [19]. CoCondenser is an enhanced version of Condenser that adds an unsupervised corpus-level contrastive loss to warm up the passage embedding space.
  - **BGE** [61]. BGE is a state-of-the-art model for general Chinese text embedding. This model utilizes RetroMAE for pre-training and then undergoes post-training with contrastive learning on large-scale paired data.

  As PROP, SEED, Condenser, and coCondenser have no available Chinese versions, we reproduce their work on the Chinese corpus described in section 5.1 based on their open-source training code and follow all settings provided in their paper [18, 19, 32, 33].
- **General Domain Pre-trained Models**
  - **BERT** [14] is a bi-directional Transformer based encoder that has a powerful ability on contextual text representations and achieves state-of-the-art performance on many NLP downstream tasks as well as IR tasks.
  - **RoBERTa** [31] shares the same architecture with BERT and is trained on a larger corpus through the MLM task.
  - **Chinese-BERT-WWM** [12] is a BERT-based model pre-trained with Whole Word Masking (WWM) strategy in Chinese corpora.
  - **Chinese-RoBERTa-WWM** [12] is a RoBERTa-based model pre-trained with Whole Word Masking (WWM) strategy in Chinese corpora.
  - **text-embedding-ada-002**[14], an embedding model developed by OpenAI, serves as a powerful tool for text search, text similarity, and code search. It achieves SOTA performance across various datasets such as BEIR[55], SentEval[11], etc.

  For the implementation of BERT, we use the Pytorch version BERT-base released by Google[15]. For the implementation of RoBERTa, Chinese-BERT-WWM and Chinese-RoBERTa-WWM, we directly use their models released on Huggingface[16].
- **Legal Domain Pre-trained Models**
  - **Legal-BERT** [5] is a BERT model pre-trained in the legal domain that directly uses the official BERT code in the pre-training stage.
  - **BERT-XS** [17] is a legal domain BERT model trained on the Chinese criminal document corpus.
  - **Lawformer** [59] apply Longformer[2] to initialize and train with the MLM task on the legal domain.
  - **SAILER** [26]. SAILER is a structure-aware pre-trained model for legal case representation. It utilizes the logical connections within a legal document's structure.

It's beneficial to understand the characteristics and functionalities of our selected baselines. These include the model type (Retriever or Re-ranker), the range of language support (English,

[13] As PROP and B-PROP have almost the same performance, we choose one of these two models as the baseline.
[14] https://platform.openai.com/docs/guides/embeddings
[15] https://github.com/google-research/bert
[16] https://huggingface.co/roberta-base, https://huggingface.co/hfl/chinese-bert-wwm, https://huggingface.co/hfl/chinese-roberta-wwm-ext
[17] http://zoo.thunlp.org

Table 1. A comparative overview of our selected baseline models. This table categorizes various pre-trained language models based on their type (Retriever or Re-ranker), language support (English, Chinese, or Multilingual), presence of an MLP layer (indicated by ✓ for presence and ✗ for absence), and the primary purpose of the MLP layer.

	Model Name	Language	Model Type	MLP Layer	MLP Purpose
IR PLM	PROP	En & Zh	Re-ranker	✓	Semantic Relevance
	SEED	En & Zh	Retriever	✗	-
	Condenser	En & Zh	Retriever	✗	-
	coCondenser	En & Zh	Retriever	✗	-
General PLM	BERT	En	Retriever / Re-ranker	✓	Next Sentence Prediction
	RoBERTa	En	Retriever	✗	-
	Chinese-BERT-WWM	Zh	Retriever / Re-ranker	✓	Next Sentence Prediction
	Chinese-RoBERTa-WWM	Zh	Retriever	✗	-
	text-embedding-ada-002	Multi	Retriever	✗	-
Legal PLM	Legal-BERT	En	Retriever / Re-ranker	✓	Next Sentence Prediction
	BERT-XS	Zh	Retriever / Re-ranker	✓	Next Sentence Prediction
	Lawformer	Zh	Retriever	✗	-
	SAILER	En & Zh	Retriever	✗	-
Ours	Caseformer Retriever	En & Zh	Retriever	✗	-
	Caseformer Re-ranker	En & Zh	Re-ranker	✓	Legal Relevance

Chinese, or Multilingual), the incorporation of an MLP layer, and the specific purposes of these MLP layers. To provide a clear and comprehensive view of these attributes, a summarized comparison of these models is detailed in Table 1.

5.5 Implementation Details

Our implementation details of Caseformer and other baselines are described as follows.

5.5.1 Caseformer and Baselines. For the Chinese version of Caseformer, we initialize our re-ranking model with the Chinese-BERT-WWM[18] and retriever with Chinese-RoBERTa-WWM[19]. For the Engish version of Caseformer, we initialize our re-ranking model and retrieval model respectively with the BERT-base-uncased[20] and RoBERTa[21] checkpoints from Huggingface. We set the hyperparameter λ in the FDM task to 16. For the implementation of Legal-BERT, BERT-XS, and Lawformer we use the checkpoints released by the original paper. As PROP, SEED, Condenser, and coCondenser have no available Chinese versions, we reproduce their work on Chinese corpora described in section 5.1 based on their open-source training code and follow all settings provided in their paper [18, 19, 32, 33]. For the BM25 and Querylikehood method, we use the pyserini toolkit[22] with default hyperparameters.

5.5.2 Pre-training Settings. In the LAM task, we follow the masking strategy of BERT. In the FDM and LJP tasks, we select every case from the corpus as the query case and generate positive cases and negative cases via the contrastive sampling strategy introduced in Section 4. We use the AdamW optimizer with a learning rate of $5e^{-6}$ and a warm-up ratio of 0.1. In the LAM task, we

[18]https://huggingface.co/hfl/chinese-bert-wwm
[19]https://huggingface.co/hfl/chinese-roberta-wwm-ext
[20]https://huggingface.co/bert-base-uncased
[21]https://huggingface.co/roberta-base
[22]https://github.com/castorini/pyserini

set the masking ratio as 0.15. In the LJP and FDM task, we set the maximum length of the query and document to 512 and truncate the rest. For the baseline PLMs, we follow all the pre-training settings in the original paper. For computing costs, we pre-trained our model on 8 Nvidia GeForce RTX 3090 GPUs for 120 hours.

Note that legal documents are often lengthy, with many exceeding the maximum input length of transformer-based models. To address this, we follow the common practice of truncating documents to fit within the model's input limits, as employed in prior works like SAILER [26]. While truncation may omit some content, the initial segments of legal documents typically contain enough information for measuring relevance. Our experiments demonstrate that using the first 512 tokens still yields strong performance, suggesting that these initial tokens carry sufficient semantic information for relevance assessment.

5.5.3 Fine-tuning Process. In alignment with existing works [20, 26, 50], our fine-tuning approach incorporates a contrastive learning strategy for both the retrieval and re-ranking models. For each query in the benchmark dataset, cases directly relevant to the query are defined as positive samples. We then randomly sample n negative samples from the cases deemed non-relevant, establishing a contrastive set for model training. The objective is to refine the model's ability to differentiate between relevant and irrelevant legal cases effectively.

We use the Softmax Cross Entropy Loss [1, 4, 20, 48] to optimize the models, which is defined as:

$$\begin{aligned} &\mathcal{L}(Q, s^+, N) \\ &= -\log \frac{exp(S(Q, s^+))}{exp(S(Q, s^+) + \sum_{s^- \in N} exp(S(Q, s^-))}, \end{aligned} \tag{17}$$

where S is the relevance score function which is defined in Equation 12 for retrieval models and is defined in Equation 9 for re-ranking models. Q is the query case, s^+ is the relevant case and N is the set of irrelevant cases randomly sampled from the corpus.

For the COLIEE and CAIL-SCM datasets, we use the Relevance Label provided in the official training set for training, and the Relevance Label in the testing set for evaluation. For the LeCaRD and CAIL-LCR datasets, since the complete datasets are relatively small (each having only about 100 queries), to ensure the robustness of the experimental results, we adopt five-fold cross-validation for fine-tuning. The dataset is divided into five equally sized subsets. In each iteration, one fold is reserved as the test set, while the remaining four folds are combined and used as the training set. This process is repeated five times, with each fold serving exactly once as the test set. The results from all five iterations are then averaged to provide a comprehensive evaluation of the model's performance.

5.5.4 Statistical Significance Evaluation. For the significance test, we adopt Fisher's randomization test [3, 10, 17] which is recommended for IR evaluation by previous work [46].

6 EXPERIMENTAL RESULTS

6.1 Caseformer Retriever

The performance of Caseformer Retriever and other baselines on LeCaRD and CAIL-LCR datasets are reported in Table 2. To be specific, we adopt recall@k (R@k) as the evaluation metric to test how many cases with labels[23] 2 and 3 are recalled by the retrieval model in the top-k results from the whole corpus. We evaluate the retrieval performance in both zero-shot and fine-tuning settings.

[23]Both LeCaRD and CAIL-LCR datasets adopt the multi-level label (0,1,2,3) to measure the relevance between the query case and candidate cases. Labels 2 and 3 indicate that the query case is strongly relevant to the candidate case.

Table 2. The experimental results of the Caseformer retriever (dual-encoder architecture) and other baselines on LeCaRD and CAIL-LCR in the zero-shot setting. The best results are in bold. "*" denotes the result is significantly worse than Caseformer with $p < 0.01$ level. R@k indicates the Recall@k metric in this table.

	Zero-shot					
	LeCaRD			**CAIL-LCR**		
	R@100	R@200	R@500	R@100	R@200	R@500
BM25	**0.5154**	**0.6781**	0.8249	0.4484*	0.6029*	0.7924
BERT	0.1538*	0.2216*	0.3351*	0.1784*	0.2475*	0.3448*
RoBERTa	0.4912	0.6011*	0.7537*	0.5619	0.6889*	0.8092*
BGE	0.4874	0.6091*	0.7533*	0.4932*	0.6191*	0.7835*
SEED	0.3311*	0.4389*	0.6378*	0.4534*	0.5721*	0.7271*
Condenser	0.3728*	0.5227*	0.6944*	0.4523*	0.5694*	0.7243*
coCondenser	0.4098*	0.5515*	0.7240*	0.4505*	0.5782*	0.7423*
text-embedding-ada-002	0.3257*	0.4232*	0.5588*	0.3777*	0.4856*	0.6479*
BERT-XS	0.1643*	0.2344*	0.3474*	0.1128*	0.1697*	0.2484*
Lawformer	0.3002*	0.3853*	0.4913*	0.3624*	0.4559*	0.5503*
SAILER	0.3731*	0.5626*	0.8148*	0.4932*	0.6537*	0.7945*
Caseformer (ours)	0.4929	0.6541	**0.8323**	**0.5648**	**0.7117**	**0.8374**

Table 3. The experimental results of the Caseformer retriever (dual-encoder architecture) and other baselines on LeCaRD and CAIL-LCR after finetuning. The best results are in bold. "*" denotes the result is significantly worse than Caseformer with $p < 0.01$ level. R@k indicates the Recall@k metric in this table. As we cannot fine-tune OpenAI models, the results for text-embedding-ada-002 in our table represent zero-shot performance.

	Fine-tuned					
	LeCaRD			**CAIL-LCR**		
	R@100	R@200	R@500	R@100	R@200	R@500
BM25	0.5154*	0.6781*	0.8249*	0.4484*	0.6029*	0.7924*
BERT	0.5292*	0.7067*	0.8506*	0.8299*	0.9189*	0.9701*
RoBERTa	0.5825*	0.7169*	0.8692*	0.8361*	0.9232*	0.9741*
BGE	0.5739*	0.6916*	0.8380*	0.8372*	0.9277*	0.9753*
SEED	0.5634*	0.7197*	0.8640*	0.8356*	0.9243*	0.9724*
Condenser	0.5937*	0.7396*	0.8666*	0.8415*	0.9301*	0.9774*
coCondenser	0.5946*	0.7425*	0.8710*	0.8403*	0.9285*	0.9766*
text-embedding-ada-002	0.3257*	0.4232*	0.5588*	0.3777*	0.4856*	0.6479*
BERT-XS	0.1769*	0.2842*	0.4368*	0.1993*	0.2758*	0.4014*
Lawformer	0.4806*	0.6465*	0.8198*	0.8139*	0.9104*	0.9637*
SAILER	0.5937*	0.7310*	0.8714*	0.8404*	0.9265*	0.9786*
Caseformer (ours)	**0.6111**	**0.7618**	**0.8958**	**0.8479**	**0.9360**	**0.9801**

Table 4. The experimental results of Caseformer and other baselines on COLIEE 2020 Task 1. The best results are in bold. "*" denotes the result is significantly worse than Caseformer with $p < 0.01$ level. R@k indicates the Recall@k metric in this table. P@5 and R@5 represent Precision@5 and Recall@5, respectively.

		COLIEE 2020				
		P@5	**R@5**	**F1**	**MRR@10**	**MRR@50**
Lexical	**BM25**	0.4754*	0.5721*	0.5192*	0.7875*	0.7907*
	QL	0.4554*	0.5506*	0.4985*	0.7906*	0.7934*
General PLMs	**BERT**	0.4542*	0.5588*	0.5011*	0.7923*	0.7948*
	RoBERTa	0.4639*	0.5862*	0.5155*	0.7613*	0.7635*
IR PLMs	**Condenser**	0.4862*	0.6127*	0.5421*	0.8198*	0.8213*
	coCondenser	0.5000*	0.6287*	0.5570*	0.8337*	0.8347*
	SEED	0.5308*	0.6952*	0.6019*	0.8683*	0.8699*
Legal Domain	**Legal-BERT**	0.4262*	0.5544*	0.4817*	0.7571*	0.7594*
	SAILER	**0.5446**	0.7152	0.6164	0.8823	0.8831
	Caseformer	0.5440	**0.7234**	**0.6210**	**0.8856**	**0.8872**

For the zero-shot setting, we directly use the model after pre-training without fine-tuning. For the fine-tuning setting, we adopt five-fold cross-validation to fine-tune the PLMs. Note that in the manual annotation stage of the LeCaRD and CAIL-LCR dataset, all the candidate cases are recalled by lexical matching methods including TF-IDF, BM25, and Query Likelihood [36]. Therefore, the annotation results have a strong bias towards lexical matching models such as BM25 in these two datasets.

Through the experimental results, we have the following observations: (1) Caseformer outperforms the traditional retrieval method BM25 in most settings on both datasets, even though the BM25 method has a strong bias on both datasets. Besides, compared with previous SOTA pre-trained language models (PLMs), Caseformer has the best performance in both zero-shot and fine-tuning settings on both datasets. These results demonstrate that Caseformer can better capture the relevance between legal cases which indicates the effectiveness of our pre-training tasks. (2) Caseformer has a more significant advantage over other PLMs in the zero-shot setting compared with the fine-tuning setting. As the zero-shot performance directly reflects the effectiveness of the pre-training tasks, this shows that Caseformer obtains more legal knowledge in the pre-training stage compared with other PLMs. (3) As lexical matching methods do not require training, BM25 performs better than most pre-trained models in the zero-shot setting. However, the performance of PLMs exceeds BM25 after fine-tuning. Despite the outstanding performance of the text-embedding-ada-002 model in general retrieval tasks, it falls short in the Legal Case Retrieval task compared to BM25. (4) The pre-trained models tailored for open-domain retrieval (e.g., SEED, coCondenser) have no significant advantages over BERT and RoBERTa. The performance of legal domain PLMs (BERT-XS, Lawformer) is lower than general domain PLMs in both zero-shot and fine-tuning settings. Showing that simply inheriting the pre-training tasks in the open domain without considering the unique characteristics of the legal field has limited benefit for case retrieval.

6.2 Multilingual Pre-training

To test the generality of our approach, we apply the Caseformer retriever to English corpora. For the pre-training dataset, we adopt the Indian Legal Statute Identification (ILSI) [40] dataset. For

the downstream dataset, we adopt COLIEE [41]. The experimental result of Caseformer and other baselines are shown in Table 4. We can observe that Caseformer outperforms lexical matching methods, general domain PLMs, PLMs for IR, and existing legal domain PLMs in English corpora. The powerful performance of Caseformer in different languages indicates that our proposed pre-training framework is language-independent and has strong generalization ability.

For reference, our F1 score of 0.6210 places us approximately in the top 37% among all participating teams in the COLIEE 2020 competition. While some top-ranking systems employed ensembles of multiple information retrieval approaches or incorporated external resources and legal knowledge, our single-model approach demonstrates competitive performance without relying on such techniques. For a detailed comparison of the participating teams' performances, please refer to the official COLIEE 2020 paper [41].

We observe that Caseformer exhibits superior performance on the Chinese datasets LeCaRD and CAIL compared to its performance on the English dataset COLIEE 2020. This disparity can be primarily attributed to the higher consistency between the pre-training corpus and the test datasets in Chinese compared to English. Specifically, the Chinese version of Caseformer is trained on legal cases from China, which maintains a consistent format across various regions within the country. This uniformity likely enhances the model's ability to generalize from training to testing conditions effectively. In contrast, the English version of Caseformer is pre-trained on legal cases from Indian courts and tested on a Canadian benchmark, the COLIEE 2020. Despite both countries employing the Case Law system and the documents being written in English, there are noticeable variations in national legal provisions and documentation styles between India and Canada. These differences likely contribute to the observed performance discrepancies between the Chinese and English versions of Caseformer.

6.3 Caseformer Re-ranker

Table 5. The experimental results of the Caseformer re-ranker (cross-encoder architecture) and other baselines on LeCaRD and CAIL-LCR in the zero-shot setting. The best results are in bold. "*" denotes the result is significantly worse than Caseformer with $p < 0.01$ level. N@k indicates the NDCG@k metric in this table.

	Zero-shot					
	LeCaRD			**CAIL-LCR**		
	N@5	N@10	N@15	N@5	N@10	N@15
BM25	0.6843*	0.7082*	0.7303*	0.7105*	0.7303*	0.7490*
QL	0.6906*	0.7168*	0.7411*	0.7389*	0.7535*	0.7756*
BERT	0.6195*	0.6293*	0.6487*	0.6183*	0.6164*	0.6259*
PROP	0.5789*	0.5982*	0.6044*	0.5823*	0.5993*	0.6090*
BERT-XS	0.6485*	0.6646*	0.6621*	0.6631*	0.6790*	0.6970*
Caseformer (ours)	**0.7831**	**0.8014**	**0.8065**	**0.8288**	**0.8330**	**0.8354**

The performance of the Caseformer re-ranker and other baselines on LeCaRD and CAIL-LCR datasets are shown in Table 5. For the zero-shot setting, we directly use the model after pre-training without fine-tuning. For the fine-tuning setting, we adopt five-fold cross-validation to fine-tune the pre-trained language models (PLMs) on each dataset. The results are shown in Table 5 and Table 7 Through the experimental results, we have the following observations:

Table 6. The experimental results of the Caseformer re-ranker (cross-encoder architecture) and other baselines on LeCaRD and CAIL-LCR after finetuning. The best results are in bold. "*" denotes the result is significantly worse than Caseformer with $p < 0.01$ level. N@k indicates the NDCG@k metric in this table. As we cannot fine-tune OpenAI models, the results for text-embedding-ada-002 in our table represent zero-shot performance.

	Fine-tuned					
	LeCaRD			**CAIL-LCR**		
	N@5	N@10	N@15	N@5	N@10	N@15
BM25	0.6843*	0.7082*	0.7303*	0.7105*	0.7303*	0.7490*
QL	0.6906*	0.7168*	0.7411*	0.7389*	0.7535*	0.7756*
BERT	0.7553*	0.7697*	0.7966*	0.7993*	0.8064*	0.8085*
PROP	0.7513*	0.7563*	0.7892*	0.7924*	0.8017*	0.8032*
BERT-XS	0.7486*	0.7668*	0.7908*	0.7828*	0.7973*	0.8126*
Caseformer (ours)	**0.8345**	**0.8357**	**0.8394**	**0.8362**	**0.8413**	**0.8433**

In case re-ranking tasks, the experimental results reveal that Caseformer achieves superior performance over all the baselines in both zero-shot and fine-tuning settings, across both datasets. After a thorough analysis, we propose two primary factors that contribute to the exceptional performance of Caseformer. Firstly, The pre-training tasks of Caseformer are specially designed for the legal field. Compared with pre-training tasks tailored for open-domain retrieval tasks (e.g., Web search), Caseformer consider the unique characteristics of the legal field, including the structured information of legal documents and the definition of relevance in the legal field. As a result, Caseformer learns legal knowledge and relevance-matching knowledge in the pre-training stage which is useful for case retrieval tasks. Also, the proposed LJP and FDM tasks can effectively teach Caseformer to measure the case relevance based on fact description and legal judgments which resembles how legal experts annotate the relevance between case documents.

In the zero-shot setting, traditional methods such as BM25 and QL outperform general domain pre-trained models (BERT), pre-trained models tailored for IR (PROP), and legal domain pre-trained models (BERT-XS). Showing that traditional methods are still strong baselines in the zero-shot setting. An interesting observation is that, despite its huge parameter size and outstanding performance in many open-domain tasks, the text-embedding-ada-002 model provided by OpenAI is still outperformed by simple lexical methods in Legal Case Retrieval. Caseformer is the only pre-trained language model that outperforms BM25 and QL in the zero-shot setting. This indicates the potential of domain-specific pre-training tasks for legal case retrieval.

To further evaluate the ability of Caseformer, we evaluate the performance of Caseformer and other baselines on CAIL-SCM, a legal case similarity judgment dataset of the Chinese Law System. The task of CAIL-SCM is to predict which case of two candidate cases is more similar to the query case and adopt the accuracy metric to evaluate the performance. In our experiment, we tested Caseformer and three types of re-rankers including a general domain pre-trained model (BERT), a pre-trained model tailored for IR (PROP), and a legal domain pre-trained model (BERT-XS).

The experimental result is shown in Table 7. We can see that Caseformer outperforms all the baselines in both zero-shot and fine-tuning settings on both test and valid datasets. Indicates that Caseformer has a strong relevance-matching ability between legal cases. Showing the effectiveness of our pre-training framework.

Table 7. The experimental results of Caseformer re-ranker and other baselines on CAIL-SCM. The best results are in bold."*" denotes the result is significantly worse than Caseformer with $p < 0.05$ level.

		CAIL-SCM	
		Valid Set Accuracy	**Test Set Accuracy**
Zero-shot	**BERT**	0.5040*	0.5149*
	BERT-XS	0.5147*	0.5124*
	PROP	0.5127*	0.5091*
	Caseformer	**0.5593**	**0.5494**
Fine-tuned	**BERT**	0.6153*	0.6393*
	BERT-XS	0.6207*	0.6517*
	PROP	0.6047*	0.6237*
	Caseformer	**0.6613**	**0.6959**

Table 8. The experimental results of ablation study on LeCaRD in the zero-shot setting. The best results are in bold. "*" denotes the performance is significantly better than the backbone model (BERT) with $p < 0.01$ level.

	LeCaRD Zero-shot		
	NDCG@5	**NDCG@10**	**NDCG@15**
only LAM	0.6531*	0.6614*	0.6690*
only FDM	0.7065*	0.7137*	0.7215*
only LJP	0.7456*	0.7498*	0.7503*
w/o LAM	0.7542*	0.7623*	0.7725*
w/o FDM	0.7513*	0.7537*	0.7582*
w/o LJP	0.7112*	0.7263*	0.7426*
Before Pre-training	0.6195	0.6293	0.6487
Caseformer(Full)	**0.7831***	**0.8014***	**0.8065***

6.4 Ablation Study

Our proposed pre-training framework caseformer contains three pre-training tasks. To analyze the influence and effectiveness of each task, we investigate all possible combinations of loss functions for three tasks and evaluate their performance on the LeCaRD dataset under the zero-shot setting. As we use BERT to initialize our model, we also provide the result of BERT for comparison.

The experimental results are shown in Table 8. We have the following findings. Firstly, each individual task contributes to the overall enhancement of the initial model's performance. Pre-training involving all three tasks leads to the highest performance while removing any task results in a decline in model performance. Secondly, removing the Legal Judgment Prediction (LJP) task leads to the most substantial performance degradation. This shows that measuring the legal similarity between cases is important for the case retrieval task and the model obtains the ability to measure the legal similarity through the LJP task. Finally, removing the LAM task leads to a relatively small degradation which shows that modeling the legal language is useful but limited compared with the LJP task and the FDM task.

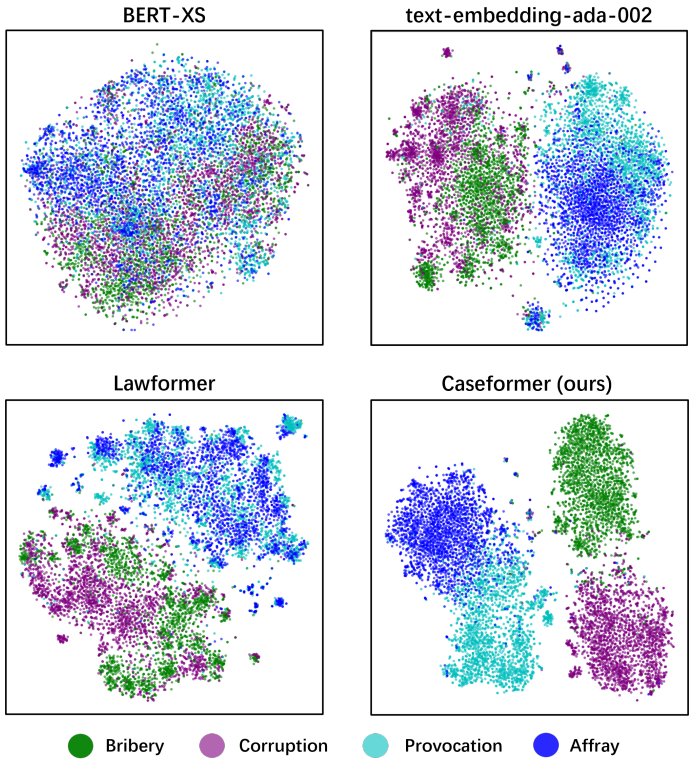

Fig. 7. Visualization results of case embeddings generated by four retrieval models.

We also conduct ablation studies on the different weights of the losses from the FDM task and the LJP task. Assigning appropriate weights to these loss components is important to balance their contributions during training. To investigate the impact of different weight settings, we conducted experiments on the Caseformer Re-ranker using various weight ratios between the FDM and LJP tasks. Specifically, we tested the ratios of 1:1, 1:2, 1:4, 2:1, and 4:1. Due to the computational demands of pre-training on our large corpus of 800 million pseudo-query-case pairs, we limited our experiments to these representative ratios.

Our findings indicate that the models trained with different weight settings did not exhibit significant differences in performance. The variations in nDCG@5, nDCG@10, and nDCG@15 across the different settings were minimal. The p-values from t-test significance tests comparing the performance of these models were greater than 0.1, suggesting that the differences were not statistically significant. This result is due to our sufficiently large pre-training corpus, which provides abundant learning signals for both tasks. Even when the loss for one task is weighted less, the model still effectively learns the objectives of both tasks. Based on these observations, we select the 1:1 weight setting for simplicity and computational efficiency in our final model.

6.5 Visual Analysis

To figure out the difference in the retrieval mechanism behind Caseformer and other baselines, we use t-SNE [56] as the dimension reduction method to visualize the case embeddings of different crimes. T-SNE is a nonlinear dimension reduction algorithm used to reduce the dimension of

high-dimensional vectors to a lower dimension. The vectors that are close to each other will remain close after the t-SNE dimension reduction.

Given a query case, retrieval methods aim to recall the cases that are close to the query in the embedding vector's distance. As a result, the visualization of case embeddings can intuitively show how the model measures the relevance between cases. Specifically, we visualize the legal cases of four crimes: Bribery, Corruption, Provocation (crime of picking quarrels and provoking trouble), and Affray. We randomly select 2500 cases for each crime and visualize the embeddings generated by different retrieval models in the zero-shot setting, which is shown in Figure 7. Note that in Chinese criminal law, the clauses of Bribery and Corruption are two different articles but share the same *category charge*[24] (the category charge of Graft and Bribery). The clauses of Provocation and Affray also share the same category charge (the category charge of Disrupting the Order of Social Administration). The cases of the same category charge are usually considered more difficult to be distinguished.

Based on the visualization result, we have the following observations. First, BERT-XS mixes different crimes showing that BERT-XS is not able to measure the similarity of cases at the legal level. Second, text-embedding-ada-002 and Lawformer divide all cases into two categories according to the category charge. This shows that text-embedding-ada-002 and Lawformer can preliminarily measure the similarity between cases at the legal level but not accurately. They can only distinguish different categories but not different charges under the same category. Finally, Caseformer divided all cases into four categories based on the crime. This indicates that Caseformer can distinguish different crimes under the same category. Compared with existing PLMs like Lawformer and text-embedding-ada-002, Caseformer can measure the similarity between cases at the legal level more precisely.

In summary, compared with other baselines, Caseformer can measure the legal similarity between cases more precisely in the zero-shot setting which indicates the effectiveness of our proposed pre-training framework.

7 LIMITATIONS AND FUTURE WORKS

In this paper, we propose Caseformer, a pre-training framework tailored for legal case retrieval that achieves state-of-the-art performance in zero-shot settings and fine-tuning with full-scale data. In this framework, we propose three pre-training objectives that enable PLMs to learn massive legal knowledge and obtain relevance-matching ability in the legal field. Extensive experiments show the effectiveness of Caseformer.

While our proposed framework has demonstrated promising results in legal case retrieval, it is important to acknowledge its limitations and outline potential directions for future research. A primary limitation of our current approach is the handling of lengthy legal documents. Legal texts often exceed the maximum input length of transformer-based models, which in our experiments is set to 512 tokens. To manage this, we have employed a truncation strategy, processing only the initial segment of each document. While this method follows common practices in the field (e.g., SAILER [26]), it inevitably omits some portions of the text. This truncation could potentially impact the model's ability to fully comprehend and assess the relevance of legal cases, especially when significant information is located beyond the initial tokens.

To address these limitations, we propose several avenues for future research:

(1) **Exploring Transformer Architectures for Longer Sequences:** Recent advancements in large language models (LLMs) have led to architectures capable of processing significantly longer input sequences. Models such as LLaMA3 support context lengths of 8k tokens, while

[24]Category charge is the general name of a certain type of crime.

other models such as Qwen2 [62] have been developed to process inputs with context lengths up to 32,000 tokens. By leveraging these LLMs, we can process larger portions or even entire lengthy legal documents, capturing comprehensive information that was previously omitted due to input length limitations.

(2) **Implementing Hierarchical Modeling Approaches:** Hierarchical modeling strategies process documents at multiple levels, capturing both local and global contextual information. By dividing a legal document into sections or paragraphs and generating embeddings for each segment, we can aggregate these representations to form a comprehensive understanding of the entire text. This approach can help in utilizing information from all parts of the document while keeping computational demands manageable.

(3) **Developing More Effective Chunking Strategies:** Chunking the document into smaller, coherent segments and employing attention mechanisms to relate information across these chunks is another promising strategy. Moreover, techniques such as recurrent neural networks or graph-based models can also be explored to enable the model to process and integrate information from different parts of a long document effectively.

(4) **Integrating Caseformer into Retrieval-Augmented Generation Frameworks:** Another promising direction is to explore how Caseformer can be applied within Retrieval-Augmented Generation (RAG) frameworks to enhance large language models [21, 23, 25, 52–54, 58]. By incorporating Caseformer as a retrieval component that supplies relevant legal cases, we can improve the factual accuracy and legal reasoning abilities of LLMs in tasks such as legal document generation, legal document summarization, and judgment prediction.

By pursuing these future directions, we aim to address the current limitations of Caseformer and expand its applicability in the legal domain. Enhancing the model's ability to process lengthy documents and integrating it with advanced generation frameworks will contribute to more effective and comprehensive legal AI systems. We believe that these efforts will significantly advance the field of legal case retrieval and support the development of tools that better assist legal professionals in their work.

ACKNOWLEDGMENTS

This work is supported by the Natural Science Foundation of China (Grant No. 61732008, 62002194) and Tsinghua University Guoqiang Research Institute.

REFERENCES

[1] Qingyao Ai, Keping Bi, Jiafeng Guo, and W Bruce Croft. 2018. Learning a deep listwise context model for ranking refinement. In *The 41st international ACM SIGIR conference on research & development in information retrieval*. 135–144.

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

[3] George EP Box, William H Hunter, Stuart Hunter, et al. 1978. *Statistics for experimenters*. Vol. 664. John Wiley and sons New York.

[4] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*. 129–136.

[5] 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).

[6] Wei-Cheng Chang, Felix X Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval. *arXiv preprint arXiv:2002.03932* (2020).

[7] Jia Chen, Haitao Li, Weihang Su, Qingyao Ai, and Yiqun Liu. 2023. THUIR at WSDM Cup 2023 Task 1: Unbiased Learning to Rank. *arXiv preprint arXiv:2304.12650* (2023).

[8] Jia Chen, Yiqun Liu, Yan Fang, Jiaxin Mao, Hui Fang, Shenghao Yang, Xiaohui Xie, Min Zhang, and Shaoping Ma. 2022. Axiomatically Regularized Pre-training for Ad hoc Search. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1524–1534.

[9] Xuesong Chen, Ziyi Ye, Xiaohui Xie, Yiqun Liu, Xiaorong Gao, Weihang Su, Shuqi Zhu, Yike Sun, Min Zhang, and Shaoping Ma. 2022. Web search via an efficient and effective brain-machine interface. In *Proceedings of the fifteenth ACM international conference on web search and data mining*. 1569–1572.

[10] Paul R Cohen. 1995. *Empirical methods for artificial intelligence*. Vol. 139. MIT press Cambridge, MA.

[11] Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449* (2018).

[12] Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. 2021. Pre-training with whole word masking for chinese bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29 (2021), 3504–3514.

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

[14] 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).

[15] Yixing Fan, Xiaohui Xie, Yinqiong Cai, Jia Chen, Xinyu Ma, Xiangsheng Li, Ruqing Zhang, Jiafeng Guo, and Yiqun Liu. 2021. Pre-training Methods in Information Retrieval. *arXiv preprint arXiv:2111.13853* (2021).

[16] Yan Fang, Jingtao Zhan, Qingyao Ai, Jiaxin Mao, Weihang Su, Jia Chen, and Yiqun Liu. 2024. Scaling Laws For Dense Retrieval. *arXiv preprint arXiv:2403.18684* (2024).

[17] Ronald Aylmer Fisher. 1936. Design of experiments. *British Medical Journal* 1, 3923 (1936), 554.

[18] Luyu Gao and Jamie Callan. 2021. Condenser: a pre-training architecture for dense retrieval. *arXiv preprint arXiv:2104.08253* (2021).

[19] Luyu Gao and Jamie Callan. 2021. Unsupervised corpus aware language model pre-training for dense passage retrieval. *arXiv preprint arXiv:2108.05540* (2021).

[20] Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. Rethink training of BERT rerankers in multi-stage retrieval pipeline. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part II 43*. Springer, 280–286.

[21] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997* (2023).

[22] Yu Guo, Zhengyi Ma, Jiaxin Mao, Hongjin Qian, Xinyu Zhang, Hao Jiang, Zhao Cao, and Zhicheng Dou. 2022. Webformer: Pre-training with Web Pages for Information Retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1502–1512.

[23] Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. *arXiv preprint arXiv:2305.06983* (2023).

[24] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906* (2020).

[25] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.

[26] 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).

[27] Haitao Li, Qingyao Ai, Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Zheng Liu, and Zhao Cao. 2023. Constructing Tree-based Index for Efficient and Effective Dense Retrieval. *arXiv preprint arXiv:2304.11943* (2023).

[28] Haitao Li, Jia Chen, Weihang Su, Qingyao Ai, and Yiqun Liu. 2023. Towards Better Web Search Performance: Pre-training, Fine-tuning and Learning to Rank. *arXiv preprint arXiv:2303.04710* (2023).

[29] 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).

[30] 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).

[31] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[32] Shuqi Lu, Di He, Chenyan Xiong, Guolin Ke, Waleed Malik, Zhicheng Dou, Paul Bennett, Tie-Yan Liu, and Arnold Overwijk. 2021. Less is more: Pretrain a strong Siamese encoder for dense text retrieval using a weak decoder. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 2780–2791.

[33] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Xiang Ji, and Xueqi Cheng. 2021. Prop: Pre-training with representative words prediction for ad-hoc retrieval. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 283–291.

[34] Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, Yingyan Li, and Xueqi Cheng. 2021. B-PROP: bootstrapped pre-training with representative words prediction for ad-hoc retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1513–1522.

[35] Yixiao Ma, Yunqiu Shao, Bulou Liu, Yiqun Liu, Min Zhang, and Shaoping Ma. 2021. Retrieving legal cases from a large-scale candidate corpus. *Proceedings of the Eighth International Competition on Legal Information Extraction/Entailment, COLIEE2021* (2021).

[36] 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.

[37] Yixiao Ma, Yueyue Wu, Weihang Su, Qingyao Ai, and Yiqun Liu. 2023. CaseEncoder: A Knowledge-enhanced Pre-trained Model for Legal Case Encoding. *arXiv preprint arXiv:2305.05393* (2023).

[38] Zhengyi Ma, Zhicheng Dou, Wei Xu, Xinyu Zhang, Hao Jiang, Zhao Cao, and Ji-Rong Wen. 2021. Pre-training for Ad-hoc Retrieval: Hyperlink is Also You Need. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1212–1221.

[39] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. *arXiv preprint arXiv:1901.04085* (2019).

[40] Shounak Paul, Pawan Goyal, and Saptarshi Ghosh. 2022. LeSICiN: A heterogeneous graph-based approach for automatic legal statute identification from Indian legal documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 11139–11146.

[41] Juliano Rabelo, Mi-Young Kim, Randy Goebel, Masaharu Yoshioka, Yoshinobu Kano, and Ken Satoh. 2021. COLIEE 2020: methods for legal document retrieval and entailment. In *New Frontiers in Artificial Intelligence: JSAI-isAI 2020 Workshops, JURISIN, LENLS 2020 Workshops, Virtual Event, November 15–17, 2020, Revised Selected Papers 12*. Springer, 196–210.

[42] Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, Vol. 242. Citeseer, 29–48.

[43] 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.

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

[45] 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.

[46] Mark D Smucker, James Allan, and Ben Carterette. 2007. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. 623–632.

[47] Weihang Su, Qingyao Ai, Xiangsheng Li, Jia Chen, Yiqun Liu, Xiaolong Wu, and Shengluan Hou. 2023. Wikiformer: Pre-training with Structured Information of Wikipedia for Ad-hoc Retrieval. *arXiv preprint arXiv:2312.10661* (2023).

[48] Weihang Su, Qingyao Ai, Xiangsheng Li, Jia Chen, Yiqun Liu, Xiaolong Wu, and Shengluan Hou. 2024. Wikiformer: Pre-training with structured information of wikipedia for ad-hoc retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 19026–19034.

[49] Weihang Su, Qingyao Ai, Yueyue Wu, Yixiao Ma, Haitao Li, and Yiqun Liu. 2023. Caseformer: Pre-training for Legal Case Retrieval. *arXiv preprint arXiv:2311.00333* (2023).
[50] Weihang Su, Yiran Hu, Anzhe Xie, Qingyao Ai, Zibing Que, Ning Zheng, Yun Liu, Weixing Shen, and Yiqun Liu. 2024. STARD: A Chinese Statute Retrieval Dataset with Real Queries Issued by Non-professionals. *arXiv preprint arXiv:2406.15313* (2024).
[51] Weihang Su, Xiangsheng Li, Yiqun Liu, Min Zhang, and Shaoping Ma. 2023. THUIR2 at NTCIR-16 Session Search (SS) Task. *arXiv preprint arXiv:2307.00250* (2023).
[52] Weihang Su, Yichen Tang, Qingyao Ai, Changyue Wang, Zhijing Wu, and Yiqun Liu. 2024. Mitigating Entity-Level Hallucination in Large Language Models. *arXiv preprint arXiv:2407.09417* (2024).
[53] Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. Dragin: Dynamic retrieval augmented generation based on the real-time information needs of large language models. *arXiv preprint arXiv:2403.10081* (2024).
[54] Weihang Su, Changyue Wang, Qingyao Ai, Yiran Hu, Zhijing Wu, Yujia Zhou, and Yiqun Liu. 2024. Unsupervised real-time hallucination detection based on the internal states of large language models. *arXiv preprint arXiv:2403.06448* (2024).
[55] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663* (2021).
[56] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).
[57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
[58] Changyue Wang, Weihang Su, Hu Yiran, Qingyao Ai, Yueyue Wu, Cheng Luo, Yiqun Liu, Min Zhang, and Shaoping Ma. 2024. LeKUBE: A Legal Knowledge Update BEnchmark. *arXiv preprint arXiv:2407.14192* (2024).
[59] 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.
[60] Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Tianyang Zhang, Xianpei Han, Heng Wang, Jianfeng Xu, et al. 2019. CAIL2019-SCM: A Dataset of Similar Case Matching in Legal Domain. *arXiv preprint arXiv:1911.08962* (2019).
[61] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-Pack: Packaged Resources To Advance General Chinese Embedding. arXiv:2309.07597 [cs.CL]
[62] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671* (2024).
[63] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems* 32 (2019).
[64] Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. LinkBERT: Pretraining Language Models with Document Links. *arXiv preprint arXiv:2203.15827* (2022).
[65] Ziyi Ye, Xiaohui Xie, Qingyao Ai, Yiqun Liu, Zhihong Wang, Weihang Su, and Min Zhang. 2023. Relevance Feedback with Brain Signals. *ACM Transactions on Information Systems* (2023).
[66] ChengXiang Zhai. 2008. Statistical language models for information retrieval. *Synthesis lectures on human language technologies* 1, 1 (2008), 1–141.