Check for updates

Event Grounded Criminal Court View Generation with Cooperative (Large) Language Models

Linan Yue
State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China
Hefei, China
lnyue@mail.ustc.edu.cn

Qi Liu*
State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China & Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
Hefei, China
qiliuql@ustc.edu.cn

Lili Zhao
State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China
Hefei, China
liliz@mail.ustc.edu.cn

Li Wang
ByteDance
Hangzhou, China
wl063@mail.ustc.edu.cn

Weibo Gao
State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China
Hefei, China
weibogao@mail.ustc.edu.cn

Yanqing An
State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China
Hefei, China
anyq@mail.ustc.edu.cn

ABSTRACT

With the development of legal intelligence, Criminal Court View Generation has attracted much attention as a crucial task of legal intelligence, which aims to generate concise and coherent texts that summarize case facts and provide explanations for verdicts. Existing researches explore the key information in case facts to yield the court views. Most of them employ a coarse-grained approach that partitions the facts into broad segments (e.g., verdict-related sentences) to make predictions. However, this approach fails to capture the complex details present in the case facts, such as various criminal elements and legal events. To this end, in this paper, we propose an Event Grounded Generation (EGG) method for criminal court view generation with cooperative (Large) Language Models, which introduces the fine-grained event information into the generation. Specifically, we first design a LLMs-based extraction method that can extract events in case facts without massive annotated events. Then, we incorporate the extracted events into court view generation by merging case facts and events. Besides, considering the computational burden posed by the use of LLMs in the extraction phase of EGG, we propose a LLMs-free EGG method that can eliminate the requirement for event extraction using LLMs in the inference phase. Extensive experimental results on a real-world dataset clearly validate the effectiveness of our proposed method. Code is available at https://github.com/yuelinan/Codes-of-EGG.

*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

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

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0431-4/24/07
https://doi.org/10.1145/3626772.3657698

CCS CONCEPTS

• **Applied computing** → **Law**.

KEYWORDS

Court View Generation, Event Extraction, Large Language Model

ACM Reference Format:
Linan Yue, Qi Liu, Lili Zhao, Li Wang, Weibo Gao, and Yanqing An. 2024. Event Grounded Criminal Court View Generation with Cooperative (Large) Language Models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3626772.3657698

1 INTRODUCTION

The remarkable success of deep neural networks has stimulated the exploration of legal intelligence applications [18, 20, 34, 42–44]. Among these applications, Criminal Court View Generation [36, 40] has garnered increasing attention as a foundational facet of legal intelligence. As depicted in Figure 1(a), the objective of criminal court view generation is to produce a coherent text, referred to as a court view, which serves as a concise representation of the case facts and offers an explanation for the rendered verdicts, such as charges and sentencing. The automated generation of court views has the potential to alleviate the workload of legal professionals while providing legal assistance to laymen [34, 40].

The existing approaches in the field can be categorized into two groups: domain-specific models [11, 36] and large language models (LLMs) [23, 30]. Several domain-specific models [34, 40] commonly generate court views by leveraging key information (e.g., crime circumstances [38–40]) extracted from the case facts using legal knowledge. For instance, C3VG [40], a court view generation model that has demonstrated promising results, explicitly categorizes crime circumstances in the case facts into two broad types: verdict-related circumstances and sentencing-related ones.

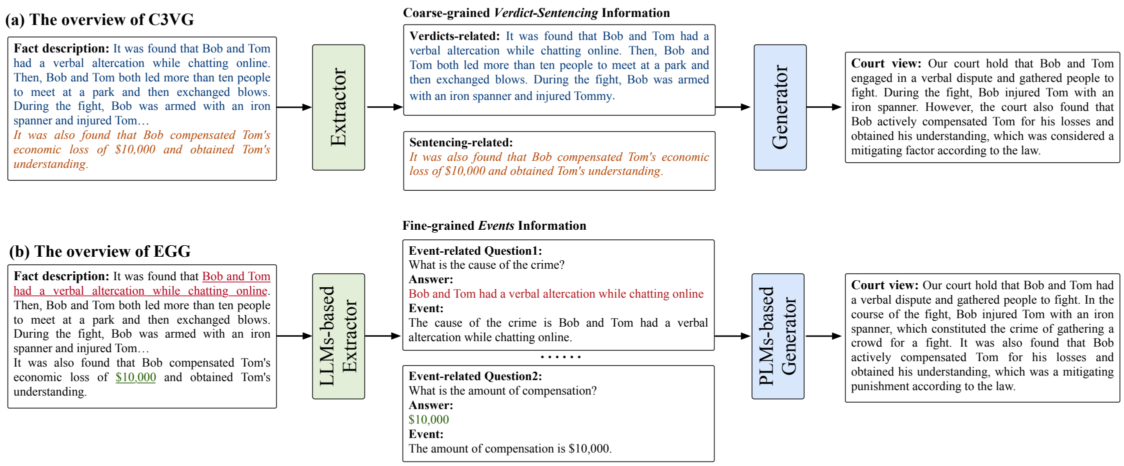

Figure 1: Schematic of C3VG and our method EGG presented in this paper.

Subsequently, it employs pre-trained language models (PLMs) (e.g., BART[1] [13]) to generate court views based on these two types of information. Nevertheless, the components comprising the case facts are highly intricate. As illustrated in Figure 1(b), case facts encompass various criminal elements[2] (i.e., legal events), represented by the underlined tokens in the fact description. Consequently, the adoption of a coarse-grained domain-specific approach that partitions the facts into two segments proves to be inadequate.

Furthermore, considering that court view generation is essentially a text generation task, it is plausible to fine-tune LLMs [37] (e.g., Baichuan-7B [2]) for court view generation. However, as evident from the experimental findings presented in Table 2, simple fine-tuning of LLMs does not yield satisfactory results. This could be attributed to the intricacy of the fact descriptions, necessitating the incorporation of additional legal knowledge. In this regard, a straightforward approach is to substitute PLMs with LLMs in domain-specific models. Nonetheless, domain-specific models often involve the collaborative training of multiple PLMs, which poses a significant computational burden on LLMs.

To this end, in this paper, we aim to develop a method which incorporates fined-grained event information into the court view generation by leveraging the collaboration between LLMs and PLMs in domain-specific models. The overview of our proposed method is present in Figure 1(b) and is two-fold: (1) extracting the fine-grained event of the case fact and (2) generating court views based on the identified events.

However, it is a non-trivial problem. Although available legal event extraction datasets contain substantial annotated data [7, 35], they primarily focus on annotating which information belongs to events in each legal document within specific case types. This approach not only necessitates extensive professional effort but also requires re-annotation of vast amounts of legal documents when encountering new case types, thereby serving as a major bottleneck for practical applications of legal event extraction. Therefore, it is crucial to devise a strategy that can extract events with minimal human annotation and demonstrate good generalization capabilities across different case types.

[1]https://github.com/yuelinan/C3VG/tree/main/bart_based_c3vg
[2]https://en.wikipedia.org/wiki/Element_(criminal_law)

To tackle the challenge mentioned above, we propose an Event Grounded Generation (EGG) method for criminal court view generation with Cooperative (Large) Language Models following an *extract-generate* framework:

- ***In the extraction phase***, we design a *LLMs*-based *event extractor*. Specifically, we first fine-tune LLMs with the publicly available legal QA dataset CJRC [6] (an extractive QA dataset like SQuAD [25]). This fine-tuning process enables the LLMs to extract pertinent answers from the original text based on a given legal question (i.e., the prompt). After the extractor is trained, we label each case type with several event-related questions. For example, as shown in Figure 1(b), for a case type of *Mobbing*, we label the event-related questions (e.g. "*What is the cause of the crime?*" and "*What is the amount of compensation?*".) Importantly, we label these questions only for the case type itself and not for individual case facts. When dealing with a specific case fact related to the crime of *Mobbing*, we utilize the pre-defined event-related questions for the *Mobbing* case type. By prompting the trained LLMs-based event extractor with the labeled questions, we extract events for each question based on the given case fact. It is worth noting that our labeled event-related questions are not present in the CJRC dataset, thus making our extraction method a zero-shot event extraction approach. Next, we combine the question and answer to get the events (e.g., the fine-grained events information in Figure 1(b)). In summary, this approach only necessitates the annotation of relevant questions for each case type, with an average of 9 questions per case type. In comparison to previous methods, our proposed event extraction approach significantly reduces the annotation time required.
- ***In the generation phase***, we splice the facts and events together to form a new text input, which is then fed into the *PLMs*-based generator to yield the court views.

Additionally, taking into consideration the computational burden posed by the use of LLMs in the extraction phase of EGG, we recognize the need to enhance its practical applicability for both laymen and professionals. To address this, we propose an LLMs-free EGG method, referred to as EGG_{free}, which eliminates the requirement of events during the inference phase. Specifically, in the training process, we still employ LLMs to extract events in the extraction phase. However, in the generation phase, instead of

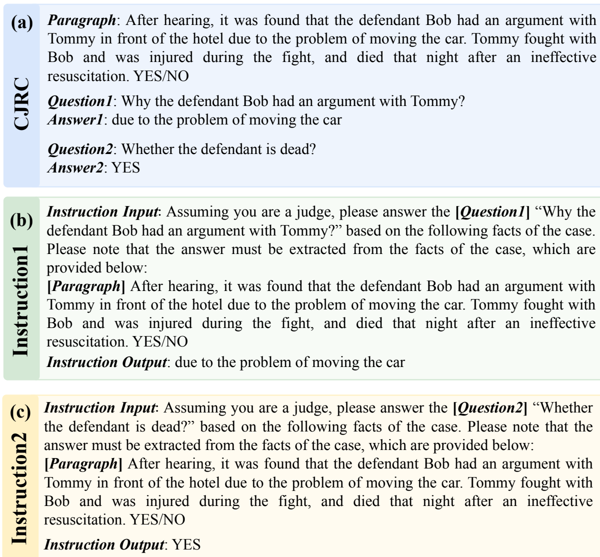

Figure 2: (a). An example from the CJRC dataset, including two questions. (b). An example from the processed instruction dataset based on the *Question1* in (a). (c). An example from the processed instruction dataset based on the *Question2* in (a).

merging the event and fact as input to the generator, we leverage the event as auxiliary information to assist the model in generating the court view based solely on the fact. To achieve this, we encode the fact and event separately using the fact and event encoders. Subsequently, we design a contrastive learning module to facilitate the fact encoder in capturing co-occurrence signals with the event through contrastive constraints. Finally, we generate the court view based on the fact. Importantly, during the inference phase, EGG$_{free}$ no longer requires any event information. It solely relies on the case fact to generate the court view without the need for event extraction. This modification aims to improve the practicality and usability of the EGG method for legal professionals and individuals without legal expertise.

In summary, the major contributions of this paper are:

- We propose an Event Grounded Generation (EGG) method for criminal court view generation with cooperative (large) language models, which *first* introduces the fine-grained event information into the court view generation.
- We propose a low data resource approach to achieve a zero-shot legal event extraction with LLMs.
- To alleviate the computational burden in EGG during inference that employs LLMs, we propose a LLMs-free EGG method based on the contrastive constraint.
- Extensive experiments on a real-world dataset validate the effectiveness of our method by comparing it with several competitive methods.

2 RELATED WORK

Court View Generation. The remarkable success in neural networks provokes the legal intelligence [3, 15, 21, 27, 43, 44]. Among them, court view generation has achieved increasing attention [36, 40]. Specifically, [36] were the first to formulate the task of court view generation and explored the use of charges to enhance the generation process, allowing the model to focus on verdict-related information within the case facts. [11] proposed a court view generation approach that involved masking key tokens in a template and subsequently employing a question-answering (QA) method to fill in these masked tokens. [34] integrated legal judgment prediction with court view generation, enabling the simultaneous generation of judgment results and court views. [40] designed an extract-generate framework that categorized case facts into two types, namely verdict-related and sentencing-related information, using an extractor. The generated court views were then based on the extracted information. Despite the promising results achieved by these existing methods, they have overlooked the incorporation of fine-grained event information present in case facts. This limitation highlights the need to consider and leverage event information for more comprehensive and accurate court view generation.

Large Language Model in Legal AI. Large Language Models (LLMs) such as ChatGPT [23] and LLaMA [30] have exhibited impressive performance across various complex tasks and have made a significant impact on society. In the realm of legal AI, researchers have been combining LLMs with legal tasks [8, 32, 41]. One notable example is Lawyer LLaMA [10], which underwent continual pre-training on an extensive legal corpus to systematically acquire legal knowledge. The model was then fine-tuned using legal instruction data, enabling it to apply its legal knowledge to specific scenarios. This approach leverages the power of LLMs to enhance the effectiveness of legal AI tasks. Another approach, ChatLaw [5], explored the use of larger base models to improve the logical reasoning capabilities of legal models. By leveraging the increased capacity and capabilities of larger models, ChatLaw aimed to enhance the model's ability to perform complex legal reasoning tasks. Privacy concerns in the legal domain are addressed by FedJudge [37], which adopts Federated Learning during the instruction tuning process. This approach ensures the privacy of legal data by training the model on local devices and only sharing aggregated updates, rather than sharing raw data. In this paper, the focus is specifically on utilizing LLMs to achieve legal event extraction.

Legal Event Extraction. In the field of legal event extraction, numerous studies [7, 14, 16, 28, 35] have involved annotating legal event types for each legal document. Notably, [35] have annotated over 8,000 legal documents with 108 event types. However, this manual annotation process is labor-intensive and time-consuming. Furthermore, when encountering new legal event types, it becomes necessary to label additional data, making existing datasets less reliable. It is crucial to develop an event extraction method that minimizes the reliance on extensive manual annotation.

3 EVENT GROUNDED GENERATION FOR CRIMINAL COURT VIEW

3.1 Problem Definition

Here, we explore the problem of criminal court view generation. We first clarify the definitions of the terms as follows:

Fact description $x = \{x_1, x_2, \ldots, x_n\}$ is the identified facts in a case including several events, where x_i denotes the i-th token.

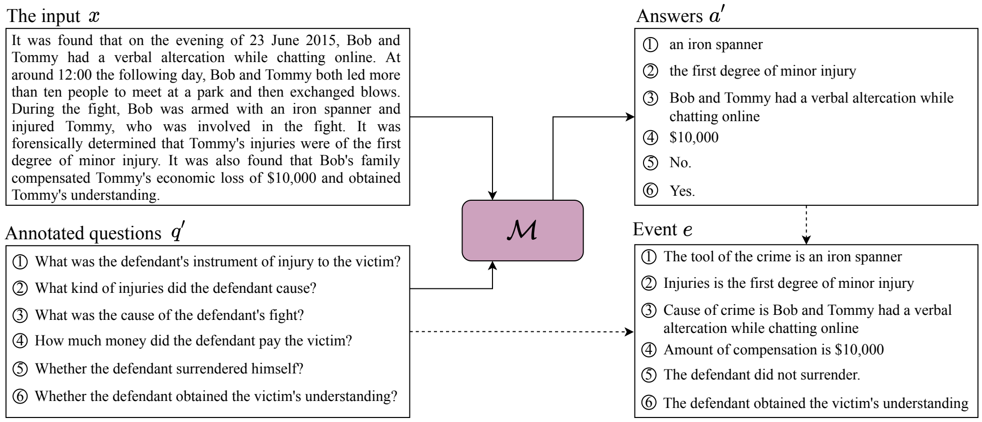

Figure 3: The process of event extraction. Among them, the input x is a legal document in the "crime of *Affray*" case type, and the questions q' is annotated for the "crime of *Affray*" case type (not only for the input x).

Event set $e = \{e_1, e_2, \ldots, e_k\}$ consists of k events of the fact, where $e_i = \{e_i^1, e_i^2, \ldots, e_i^m\}$ contains m tokens and each event is a subsequence of the fact.

Court view is the summary of the fact which consists of the charge c and rationales $r = \{r_1, r_2, \ldots, r_t\}$. Among them, the rationale is concluded from the fact in order to determine and support the judgment results, such as sentencing. In this work, we assume the charge is available, and we only focus on generating rationales in court views, where the charge can be easily obtained by the judge or the charge prediction systems [39, 42, 43].

Then, based on the above definitions, our problem is defined as:

Problem 1 (Court View Generation). *Given the case fact x, our goal is first to extract several events e from the case fact, and then generate the rationales r in court views, where the gold events are unavailable.*

3.2 Architecture of EGG

Our proposed Event Grounded Generation (EGG) for criminal court view method consists of two phases, cascading the *event extractor* and the *court view generator*. Specifically, in the extraction phase, we first train a LLMs-based QA model which can extract a subsequence of the text input as the answer to the prompts (or questions). After the model is trained, we consider this model as the *event extractor* to select several events from the case fact by introducing annotated legal event-related questions. Finally, we employ a PLMs-based *court view generator* to generate court views by merging the fact and event as the new text input.

3.2.1 ***Event extractor.*** Existing legal event extraction datasets [7, 35] mainly focus on annotating each case under different case types (i.e., different charges). However, this annotation requires significant and expensive professional labor. Meanwhile, when facing a new case type, it commonly needs to be re-labeled. To this end, we develop a zero-shot LLMs-based legal event extractor.

Specifically, we implement the *extractor* with a publicly available legal QA dataset CJRC [6], which consists of the paragraph p, question q and answer a as shown in Figure 2(a). Among them, the answer a is a part of paragraph p. Besides, to answer the question about YES or NO, CJRC adds "YES/NO" at the end of the paragraph. Based on CJRC, we train a legal LLMs-based model $\mathcal{M}$ to extract answers from the paragraph according to the questions.

In detail, we begin by transforming the original CJRC dataset into an instruction dataset [29], denoted as $\mathcal{D}$, where each instruction data has the form of $\{InstructionInput : InstructionOutput\}$. Figure 2(b) and Figure 2(c) illustrate the specific format of the prompt, task-specific instruction, and ground truth in our instruction dataset. Next, we utilize the instruction tuning method to fine-tune the base generative LLMs to extract answers from paragraphs. To address the computational and time constraints associated with directly fine-tuning the entire LLM, we employ the parameter-efficient fine-tuning technique for training the extractor. Specifically, we employ the LoRA [9] method which involves freezing the pre-trained model parameters and introducing trainable rank decomposition matrices into each layer of the Transformer architecture [31]. Finally, the learning objective can be computed as:

$$\mathcal{L}_e = -\sum_{t=1}^{|y|} \log \left(P_{\Theta+\Theta_L} \left(y_t \mid m, y_{<t} \right) \right), \tag{1}$$

where m and y represent the *Instruction Input* and *Instruction Output*, y_t denotes the t-th token of y, $y_{<t}$ is the tokens before y_t, and Θ represents the frozen LLMs parameters and Θ_L is the trainable LoRA parameters ($\Theta_L \ll \Theta$).

After the LLMs-based extractor is trained, we annotate several questions for each case type, where each question q' is related to the event in case facts x. It is important to note that we label the questions only for the case type and not for each individual case fact. For instance, if we have a case fact related to the crime of *Affray*, we utilize the previously labeled questions for *Affray*, such as the cause of the crime, tools of the crime, and whether to surrender. Then, as shown in Figure 3, we promote the trained LLM $\mathcal{M}$ to answer these event-related questions and obtain the corresponding answers a' (i.e., $a' = \mathcal{M}(x, q')$). Finally, we combine the obtained answers a' and the corresponding questions q' to obtain the event e. It is worth emphasizing that the labeled event-related questions used in this process are not present in the CJRC dataset. This characteristic distinguishes our extraction method as a zero-shot event extraction approach, as it successfully extracts event information from case

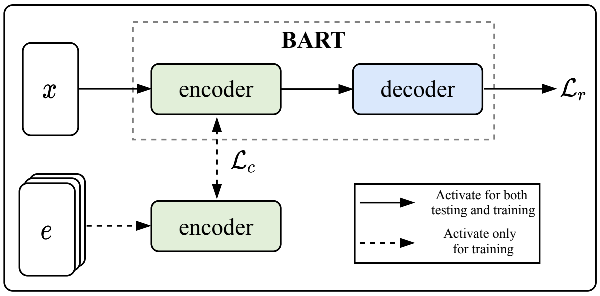

Figure 4: Architecture of EGG$_{\text{free}}$.

facts using questions that were not available in the training data. This zero-shot capability increases the versatility and adaptability of our approach to handle new or unseen case types.

3.2.2 ***Court view generator***. Previous models generate court views based solely on case fact. In this section, our *court view generator* designs a strategy to incorporate extracted event information into the fact to yield more plausible court views, where we adopt the BART [13] as our backbone by considering the advantages of the current PLMs. Specifically, we merge the event and fact descriptions to form new input x' of the *court view generator*. In practice, limited by the maximum length of the PLMs, we enforce the events to be placed before facts (i.e., $x' = e|||x$), where "|||" represents the process of mergers.

3.3 Training and Inference

In this section, we describe the training loss in our proposed method EGG. Specifically, in the extraction phase, we employ Eq(1) to train our LLMs-based event extractor. In the generation phase, we adopt the negative log-likelihood loss to optimize the generator:

$$\mathcal{L}_r = -\sum_{t=1}^{|r|} \log \left(P_{\Theta_B} \left(r_t \mid x', r_{<t} \right) \right), \tag{2}$$

where Θ_B is the trainable BART parameters, r_t denotes the t-th token of r and $r_{<t}$ is the tokens before r_t.

During the inference phase, given a description of case fact, we first use the LLMs-based *event extractor* to extract the events from the case fact. Then, we generate the court view based on both the facts and events.

4 EGG$_{\text{free}}$: LLMS-FREE EGG WITH CONTRASTIVE CONSTRAINTS

Indeed, the use of LLMs for event extraction in the extraction phase of EGG can lead to increased computational burden during the inference phase. This limitation hampers the practical application of the model in real-world scenarios. To overcome this challenge, we propose an LLMs-free EGG method that employs contrastive constraints, enabling court view generation without the need for event information during the inference phase.

4.1 Architecture of EGG$_{\text{free}}$

During training, EGG$_{free}$ follows the *extractor-generator* framework. Specifically, in the extraction phase, similar to EGG, we still employ LLM to extract events. In the generation phase, unlike the previous EGG of combining event and fact as inputs to the generator, we use event as a kind of auxiliary information to assist the model in generating court views based on fact. In particular, as shown in Figure 4, given the fact and event, we first employ the fact encoder and event encoder to encode both fact x and event e as the corresponding representations $h_x \in \mathbb{R}^d$ and $h_e \in \mathbb{R}^d$, where d is the dimensional size. Then, we feed the fact representation into the decoder for court view generation. In practice, we use the encoder and decoder of BART to achieve the above implementation.

Subsequently, to enable the fusion of event information into EGG$_{free}$, we employ a novel contrastive learning strategy during the training phase. This strategy aims to teach the fact encoder to memorize the co-occurrence event signals within its parameters, allowing the fact encoder to inject event clues into fact representations during the inference phase.

In particular, during the training phase, as shown in Figure 4, we adjust the parameters of the fact encoder based on the event encoder to maximize the mutual information between the case fact and event. To achieve this objective, for a training fact representation h_x, we build its positive sample set using its corresponding event h_e (referred to h_e^+), i.e., $\mathcal{N}^+ = \{h_e^+\}$, and its negative sample set $\mathcal{N}^- = \mathcal{N}_{batch} \backslash \mathcal{N}^+$ where $\mathcal{N}_{batch}$ denotes each batch of event samples. To teach the fact encoder to memorize the co-occurrence event signals, we define the contrastive loss following the concept of InfoNCE [22]. The contrastive loss is formulated as:

$$\mathcal{L}_c = -\mathbb{E}_{h_e^+ \in \mathcal{N}^+} \left[\log \frac{\exp\left(\frac{\text{sim}(h_x, h_e^+)}{\tau}\right)}{\exp\left(\frac{\text{sim}(h_x, h_e^+)}{\tau}\right) + \sum_{h_e^- \in \mathcal{N}^-} \exp\left(\frac{\text{sim}(h_x, h_e^-)}{\tau}\right)} \right], \tag{3}$$

where $\text{sim}(h_x, h_e)$ represents the similarity measure between the fact representation h_x and the event representation h_e, and τ is a temperature parameter that controls the sharpness of the probability distribution. Besides, we set the fact encoder and the event encoder to share parameters to save GPU memory. According to our experiments, separate encoders and shared encoders do not have a significant difference in the generation performance.

4.2 Training and Inference

In the training process, since EGG$_{free}$ uses the same extractor as EGG, we use Eq(1) to train the extractor. Besides, the final objective of the generator in EGG$_{free}$ is defined as:

$$\mathcal{L}_{\text{EGG}_{free}} = \mathcal{L}_r + \beta \mathcal{L}_c, \tag{4}$$

where β is the adjusted hyperparameter.

During the inference phase, since the fact encoder learns to capture co-occurrence signals with the event through contrastive constraints, EGG$_{free}$ can ignore the event as the input, enabling the generation of contextually relevant court views based solely on the case fact.

By leveraging contrastive constraints, our proposed method eliminates the reliance on LLMs for event extraction in the inference phase. This approach significantly reduces the computational burden, making the model more suitable for real-world applications.

Table 1: The statistics of datasets.

CJO	Results
# Sample	62,939
# Types of cases	62
# Avg. Length of fact description	458.1
# Avg. Length of court view description	130.9
# Avg. Annotated questions of event	9.0
# Avg. Length of annotated questions of event	10.8
# Avg. Length of all events in a case	83.4
CJRC	**Results**
# Sample	20,000
# Avg. Length of paragraph	501.8
# Avg. Length of question	16.5

5 EXPERIMENTS

To evaluate the effectiveness of EGG, we conduct experiments to answer the following research questions:

- **RQ1:** How effective are EGG and EGG$_{free}$ in improving the performance of event extraction and court view generation?
- **RQ2:** How efficient is EGG$_{free}$ during the inference phase?
- **RQ3:** What are the performances of EGG by the length of court views?
- **RQ4:** How do EGG and EGG$_{free}$ perform in human evaluation?
- **RQ5:** What is the court view generated by EGG to a specific case fact?

5.1 Datasets

In the extraction phase, we adopt the criminal cases in CJRC[3] [6] as the training data, where we process CJRC into the format of an instruction dataset. Figure 2 is an example from the CJRC and the instruction dataset. In the generation phase, following [40], we conduct experiments on CJO[4], where CJO is collected from the published legal documents in China Judgments Online[5]. Detailed dataset statistics are shown in Table 1. Among them, since there exist 62 types of cases, we ask three law expects to annotate questions for each case type, for a total of 558 questions.

5.2 Experimental Setup

In this section, we present the detailed experimental setup of our proposed EGG. First, in the extraction phase, we adopt Baichuan-7B [2] as the backbone of the LLMs-based event extractor $\mathcal{M}$. Then, we employ the LoRA to parameter-efficient fine-tune it on the instruction dataset. For training, we adopt an AdamW optimizer [19] with an initial learning rate of 1e-5, then we set the maximum sequence length as 512 and the batch size as 4. Besides, the rank of LoRA is set to 4. In the generation phase, we employ BART [13] to generate the court views. We set the learning rate to 1e-4 and the batch size to 8, and β in EGG$_{free}$ to 1. For evaluation, we adopt macro-average F1 as our metric to evaluate the performance of the LLMs-based event extractor $\mathcal{M}$ in the test set of CJRC. Besides, since there exist no gold events in CJO, we assume that the better the generated court views perform, the more effective events are extracted. To this end, to evaluate the performance of the generation, we adopt ROUGE [17] and BLEU [24] as the metrics. Among them, we report F1 scores of ROUGE-1, ROUGE-2, and ROUGE-L, and we keep the result of BLEU-1, BLEU-2 and BLEU-N (i.e., an average score of BLEU-1, BLEU2, BLEU-3, and BLEU-4).

5.3 Comparison methods

In this section, to evaluate the generated court view, we employ three type of baselines. First, we compare EGG with several traditional baselines:

- **AttS2S** [1] is an attention-based sequence-to-sequence model, following an encoder-decoder framework.
- **PGN** [26] employs a pointer network to solve the out of vocabulary (OOV) problem in the text generation.
- **Transformer** [31] has been widely implemented to generate texts.
- **Label-AttS2S** [36] is designed to generate court views by introducing the charge semantics into AttS2S.
- **C3VG** [40] separates the case fact into two parts with an *extract-generate* framework to generate the court views.

The above baselines are implemented with GRU [4] or transformer. For a fair comparison, the results of the above baselines are directly taken from [40].

Besides, since the pre-training models have promoted the text generation in recent years, we introduce several approaches based on the pre-training models:

- **BART** [13] is a Transformer-based pre-training sequence-to-sequence model, which achieves promising results in text generation. In this paper, BART(Fact) denotes BART takes the case fact as the input. BART(Event) represents taking the extracted event as the input.
- **C3VG with BART** [40] implements C3VG with BART as the backbone.

Finally, we also compare LLMs baselines with EGG:

- **Baichuan-7B** [2] is a large language model which achieves competitive results in Chinese intelligence tasks.
- **Baichuan-7B(Fact)** employs LoRA to fine-tune Baichuan-7B by taking the case fact as the input with the form of the instruction dataset. Among them, the *Instruction Input* is: "Assuming you are a judge, please summarize the facts of the case: [the description of case facts]", and the *Instruction Output* is the court views.

5.4 Performance on Event Extraction and Court View Generation (RQ1)

5.4.1 ***Results of event extraction.*** In this section, we report the macro-average F1 to evaluate the performance of the LLMs-based extraction model $\mathcal{M}$ in the test set of CJRC. After statistics, the macro-average F1 is **84.6** which performs better than the original macro-average F1 (82.9) reported in the paper of CJRC [6] which employs BERT [12] to achieve the answer extraction. This observation demonstrates the effectiveness of LLMs on extraction. However, our goal is employing the trained $\mathcal{M}$ to predict the potential events in our court view data CJO. Therefore, the F1 scores in the test set of CJRC fail to illustrate the effectiveness of our extraction sufficiently. To further evaluate the extracted events, we consider that the better

[3]https://github.com/china-ai-law-challenge/CAIL2019
[4]https://github.com/bigdata-ustc/C3VG
[5]https://wenshu.court.gov.cn

Table 2: Results of court view generation. "*": results obtained from C3VG [40].

Models	ROUGE (↑)			BLEU (↑)			Bert-S (↑)
	R-1	R-2	R-L	B-1	B-2	B-N	
AttS2S*	58.7	38.9	59.4	50.5	41.0	38.0	–
PGN*	59.3	37.0	59.8	50.2	39.6	36.7	–
Transformer*	59.9	39.6	60.9	50.8	41.3	38.1	–
Label-AttS2S *	47.0	31.4	52.8	38.7	31.6	29.4	–
C3VG*	60.1	40.5	62.5	52.1	43.5	40.6	–
BART(Event)	66.04	50.46	54.01	52.16	47.95	46.76	81.93
BART(Fact)	74.96	58.34	62.44	54.94	52.55	51.46	85.59
C3VG with BART	75.59	64.22	65.11	56.16	53.58	52.71	85.61
Baichuan-7B	57.43	37.76	38.65	58.96	53.92	52.01	72.84
Baichuan-7B(Fact)	74.05	60.25	61.69	**69.48**	**66.02**	**64.71**	82.58
EGG$_{free}$	75.23	64.41	64.24	57.19	54.34	53.41	85.86
EGG	**76.86**	**65.15**	**65.90**	56.92	54.43	53.59	**86.54**

Table 3: Inference speed of Baichuan-7B and EGG.

Methods	Inference Speed
Baichuan-7B(Fact)	1.00 ×
EGG$_{free}$	0.71 ×
EGG	8.72 ×

the generated court view performs, the more effective the extracted events will be. The corresponding results are shown in section 5.4.2.

5.4.2 ***Results of court view generation.*** To validate the effectiveness of EGG, we first compare it with several baselines. As shown in Table 2, we find all methods that exploit PLMs outperform the traditional baselines implemented with GRU or transformer, which demonstrates the effectiveness of PLMs. Then, we can observe that both EGG and EGG$_{free}$ perform better than other baselines in most metrics, which indicates our EGG can generate more plausible court views. Specifically, compared with C3VG with BART, which groups the original fact into two type paragraphs to generate court views, our EGG significantly outperforms it. This observation demonstrates that incorporating fine-grained events into court view generation is more effective than employing coarse-grained paragraphs. We also implement BART by taking fact and event as the text input, respectively. From the results, we observe that BART(fact) surpasses BART(Event) by a large margin, illustrating that there exist several events are not extracted. These observations prove that it is necessary to design an incorporated strategy to combine the case facts with the event information to generate court views.

Besides, we observe that Baichuan-7B without instruction tuning performs well on the legal task in the zero-shot setting, which indicates that Baichuan-7B has already possessed court view abilities through training on a large amount of data. However, its results are still worse than the fine-tuned model (Baichuan-7B(Fact)), which also shows the necessity of fine-tuning LLMs to the court view generation. Although Baichuan-7B(Fact) achieves promising results on BLEU than EGG, it still performs worse on ROUGE and Bert-S. Meanwhile, EGG does not fine-tune LLMs in the generation phase, which further illustrates the effectiveness of EGG. Finally, we analyse the difference between EGG and EGG$_{free}$. From Table 2, EGG performs better than EGG$_{free}$, which indicates that it is more effective to combine case facts and events directly and explicitly on the data side than to introduce events implicitly into the model structure. However, from the point of view of inference speed and computational resources occupied by the model, EGG$_{free}$ is faster and occupies fewer computational resources, yet achieves similar results to EGG. This observation illustrates the effectiveness of EGG$_{free}$ which employs the contrastive learning constraint to incorporate event information into the learning of factual representations. In section 5.5, we will further illustrate the efficiency of inference in EGG$_{free}$.

5.5 Efficiency of Inference in EGG$_{\text{free}}$ (RQ2)

In this section, we present the results of our experiments comparing the inference speed of our proposed EGG$_{free}$ with other baselines. The hardware setup for the experiments consists of 12 cores of Intel(R) Xeon(R) Gold 5317 CPU and a single 40G NVIDIA A100 Tensor Core GPU. The findings are summarized in Table 3. Our proposed EGG$free$ achieves an impressive decoding speed, approximately 12 times the speed achieved by EGG, which utilizes LLM for event extraction. It is worth noting that EGG has the slowest inference speeds. Besides, although the difference between the number of parameters in EGG and Baichuan-7B is not significant, since there are multiple events for a single case, EGG often needs to perform multiple event extractions, and thus is slower than Baichuan-7B. This observation highlights that EGG$free$ strikes a balance between efficiency and effectiveness, making it well-suited for resource-constrained users. The results demonstrate that EGG$_{free}$ offers a practical solution for legal event extraction, providing efficient performance while maintaining effectiveness. Its suitability for resource-constrained users makes it a valuable option in real-world applications.

5.6 Performance by the Length of Court Views (RQ3)

In this section, we focus on investigating the generation performance of court views based on their length. We sample examples from the test set of CJO, where the real court views have lengths ranging from 50 to 120 tokens. We then predict and evaluate the generated court views by comparing them with the outputs of EGG and C3VG with BART using ROUGE-L and BLEU-4 scores. The findings, as illustrated in Figure 5, reveal that both EGG and C3VG with BART experience a degradation in performance as the length of court views increases. However, we observe that our EGG achieves the best performance when the court view length is between 60

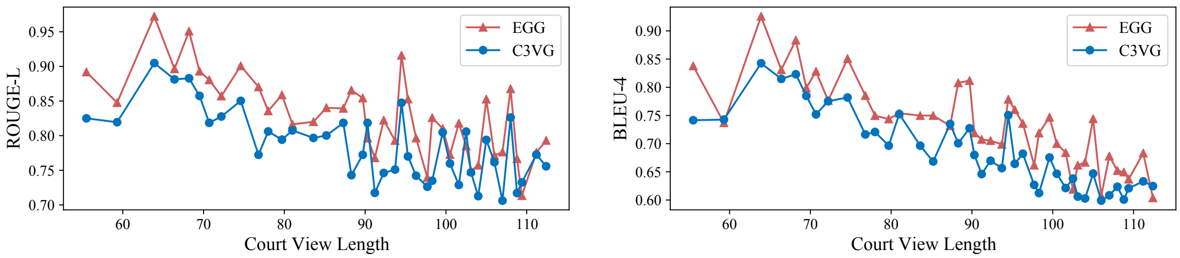

Figure 5: Model performance by the length of court views. Among them, C3VG is implemented with BART.

Table 4: Detailed scoring standards for human annotators.

Score	Usefulness	Fluency
1	No Use. The generated texts are useless for answering questions.	Nonsense.
2	Almost useless. Almost all generated texts are useless.	Very unfluent.
3	Half of them are useful. About half of the generated texts are useful for answering questions.	Partial fluent.
4	Highly useful. Most generated texts are useful to answer the questions.	Highly fluent.
5	Exactly. Generated texts are useful for me to get the correct answer.	Very fluent.

Table 5: Human evaluation on generated texts.

Methods	Usefulness	Fluency
Baichuan-7B(fact)	3.76	4.43
EGG$_{free}$	3.98	4.63
EGG	**4.09**	**4.68**

and 70 tokens, with both ROUGE-L and BLEU-4 scores surpassing 90. Furthermore, our method outperforms C3VG with BART across all court view lengths, indicating the effectiveness of incorporating fine-grained event information into court view generation. This suggests that by considering the specific event details in the generation process, our approach can produce more accurate and higher-quality court views compared to existing methods.

5.7 Human Evaluation (RQ4)

Table 2 highlights that both EGG and EGG$_{free}$ exhibit lower BLEU scores compared to Baichuan-7B(fact), prompting the need to investigate the performance of generated court views. To gain further insights, a human evaluation is conducted on the court views generated by EGG and Baichuan-7B(fact). In this evaluation, a total of 100 examples are sampled, and three annotators with expertise in both computer science and law are asked to evaluate the generated court views based on two metrics: Usefulness and Fluency. Each metric is scored on a scale from 1 (lowest) to 5 (highest), with specific scoring standards provided in Table 4. The experimental results are presented in Table 5. The results indicate that all models achieve promising scores in terms of Fluency, indicating that the generated court views are fluent and well-formed. Additionally, it is observed that EGG and EGG$_{free}$ outperform Baichuan-7B(fact) in terms of Usefulness. This finding further illustrates the effectiveness of incorporating fine-grained event information into court view generation. By considering the specific event details, our models generate court views that are deemed more useful by human evaluators.

5.8 Case Study (RQ5)

An example of extracted events and generated court views is shown in Figure 6. Specifically, firstly, the type of case in this fact is *intentional injury*. Then, **Event-related Questions** show all questions designed for the crime of intentional injury. It is worth noting that the designed questions are the same for any fact which belongs to *intentional injury*. In the **Answers**, we present the answers extracted from the fact description according to the questions by the LLMs-based extractor. Afterward, we post-process the questions and answers to obtain the corresponding **Events**.

Next, we present 7 court views generated by EGG and baselines:

• We can find although C3VG generates court views well, it fails to generate the court views about obtaining the victim's understanding. Conversely, EGG can generate more plausible court views. Besides, EGG also yields ***injured another person with a knife*** which has been described in fact description but not in the real court view. This observation indicates EGG can generate several key information, which is ignored by the real court view.

• Besides, compared to BART(Event), we find it generates several unfaithful court views (the underlined) which do not exist in the case fact, illustrating that it is unfeasible to generate court views based solely on events, and we need to combine the fact and event for the generation. Then, although BART(Fact) has generated court views well, there are several omissions it generates compared to other methods. For example, EGG yields the injuries of the defendant is minor, however, BART(Fact) only generates the defendant has injuries (i.e., ***intentionally injured another person's body***).

• Moreover, we can get a fluent court view by directly prompting Baichuan-7B (without fine-tuning). However, we tend to obtain some redundant information (e.g., "*If you do not accept this judgment, ... , directly to the [Province] Intermediate People's Court ...*"). This information has nothing to do with the court view, and may even involve some private information, such as *[Province] Intermediate People's Court*, where we have a privacy treatment for [Province] to show this example. Meanwhile, Baichuan-7B(fact) is not as accurate as EGG in yielding information about surrender, where Baichuan-7B(fact) only describes the defendant's surrender, while EGG also describes how the defendant surrendered ("*the victim learned that others had called the police and waited at the scene to be arrested*"). These observations demonstrate that incorporating fine-grained event information into the court view generation is effective.

Fact Description	The trial found that at 19:00 on June 24, 2017, the defendant Qian xx was invited by the victim Zhang xx to drink at the xx County xx Township Hotel. During that time, Qian xx and Li xx, who was at the same table, had a verbal altercation and Qian xx left the restaurant alone. The two parties agreed by telephone to meet on the highway near Qian xx's house. After the meeting, Qian xx and Zhang xx were drinking at a roadside kiosk when another argument broke out, and Zhang xx pushed Qian xx, who then used a kitchen knife he had picked up on the ground to cut Zhang xx's back and chest and cut his fingers. The extent of Zhang's injuries was assessed to be second-degree minor injuries. After learning that the police had been called, Qian xx waited at the scene to be arrested and, upon arrival, truthfully confessed to the facts of his crime. It was also found that the defendant Qian xx compensated the victim Zhang xx for damages totalling RMB 4,000, and obtained Zhang xx's understanding Yes/NO.
Event-related Questions	① What was the defendant's instrument of injury to the victim? ② What kind of injuries did the defendant cause? ③ What was the cause of the defendant's fight? ④ How much money did the defendant pay the victim? ⑤ Whether the defendant surrendered himself? ⑥ Whether the defendant obtained the victim's understanding? ⑦ Whether the defendant made a truthful confession to the crime?
Answers	① a kitchen knife ② The extent of Zhang's injuries was assessed to be second-degree minor injuries ③ another argument broke out, and Zhang xx pushed Qian xx ④ RMB 4,000 ⑤ Yes ⑥ Yes ⑦ Yes
Events	The tool of the crime is a kitchen knife. Injuries is the extent of Zhang's injuries was assessed to be second-degree minor injuries. Cause of crime is another argument broke out, and Zhang xx pushed Qian xx. Amount of compensation is RMB 4,000. The defendant surrendered himself. The defendant obtained the victim's understanding. The defendant made a truthful confession to the crime.
Real Court View (ground truth)	The court held that Qian xx intentionally injured the health of others, causing one person second-degree minor injuries, his behavior has constituted the crime of intentional injury. The defendant Qian xx was waiting for arrest at the scene knowing that others had reported the crime, and after arriving at the case, he truthfully confessed the facts of his crime, and was a self-surrender, which can be a lighter or lesser punishment according to the law. After the crime was committed, the defendant Qian xx compensated the victim for the loss and obtained an understanding, so he could be given a lighter punishment.
EGG	The court held that Qian xx intentionally ***injured another person with a knife***, causing minor injuries to one person, and his conduct constituted the crime of intentional injury. The defendant was armed with a knife to commit the act of injury, and could be punished with discretionary severity. After the crime, ***the victim learned that others had called the police and waited at the scene to be arrested***, and after arriving at the case, he truthfully confessed the facts of his crime, which could be a lighter punishment according to the law. Besides, he compensated the victim and could be punished less severely.
C3VG	The court held that the defendant Qian xx intentionally injured another person's body, causing one person minor injuries, his behavior has constituted the crime of intentional injury. The defendant Qian xx, after learning that others had called the police, waited at the scene to be arrested, and after arriving at the case, confessed truthfully to the facts of his crime, and was a self-surrender, which could be punished less severely.
BART(Event)	The court held that Qian xx, together with others, randomly assaulted others, causing minor injuries to one person, the circumstances are bad, his behavior has constituted the crime of intentional injury. ***In the joint crime, the victim, Zhang xx, was generally at fault and could be punished less severely at discretion***. The defendant confessed the facts of his crime truthfully after his return to the court, and he can be punished lightly according to the law.
BART(Fact)	The court held that Qian xx disregarded the national law and ***intentionally injured another person's body***, his behavior has constituted the crime of intentional homicide. The defendant was informed that others had called the police and waited for arrest at the scene. After arriving at the case, he truthfully confessed to the facts of his crime and was a self-surrender, which can be a lighter punishment according to the law. He compensated the victim and could be punished less severely.
Baichuan-7B	This court held that the defendant Qian xx intentional injury to the body of another person to cause minor injuries has constituted the crime of intentional injury. The public prosecution authorities accused of the facts are clear, the evidence is indeed sufficient, the charges are established. Defendant Qian xx voluntarily surrendered, truthfully confessed his crime, is the surrender, can be mitigated or reduced punishment. Defendant Qian xx actively compensated the victim's economic losses, obtained the understanding of the victim, may be appropriate to mitigate his punishment. According to the defendant's criminal circumstances and repentance, pronouncement of probation has no significant adverse impact on the community in which he lives, so according to the law to apply probation. Accordingly, in accordance with the provisions of Article 234(1), Article 67(1), and Article 72 of the Criminal Law of the People's Republic of China, the verdict is as follows Defendant Qian xx is guilty of intentional injury and sentenced to six months of fixed-term imprisonment and one year of probation. (The period of probation shall be calculated from the date when the judgment is finalized.) ***If you do not accept this judgment, you may, within ten days from the second day of receipt of the judgment, appeal through this court or directly to the [Province] Intermediate People's Court. In the case of a written appeal, one original and two copies of the statement of appeal shall be submitted.***
Baichuan-7B (Fact)	This court held that the defendant Qian xx intentionally injures another person's body to cause a minor injury, his behavior has constituted the crime of intentional injury. The public prosecution authorities accused of the facts are clear, the evidence is indeed sufficient, the crime is established. ***Defendant Qian xx voluntarily surrendered to the police*** and confessed to his crime after arriving at the crime, so he is a self-surrender, and can be mitigated or alleviated in accordance with the law. Defendant Qian。xx actively compensated the victim's financial losses, obtained the understanding of the victim, may be appropriate to mitigate punishment.

Figure 6: An example of extracted events and generated court views.

6 DISCUSSION

Ethical Discussion. Court view generation has gained significant attention as a core task in legal intelligence. Based on the experimental results, EGG demonstrates the ability to generate more accurate court views. Additionally, EGG$_{free}$ achieves a balance between modeling effectiveness and inference efficiency, making it suitable for users with limited computational resources. However, it is important to note that our model does not replace the work of judges. Instead, our aim is to assist judges in organizing court views and alleviate their workload. The final court views must be determined and decided upon by the judges themselves [33, 40]. Our work serves to provide judges with a tool to streamline the process of collating court views and reduce their workload stress.

Limitations. When extracting events, we simply post-process the extracted answers and questions to obtain the corresponding events. However, when dealing with complex relationships among events, such as causality, a more advanced approach is needed. One possible solution is to construct an event graph that represents the relationships among events. An event graph is a graphical representation where events are nodes, and the relationships between events are represented by edges. By incorporating this event graph into the court view generation process, the model can better capture and understand the complex relationships among events. We will leave it as the future work.

7 CONCLUSION

In this paper, we proposed an Event Grounded Generation (EGG) method for criminal court view generation with cooperative (Large) Language Models, cascading the event extractor and the court view generator. To be specific, EGG first employed a trained LLMs-based legal event extractor to select several events in the case fact without massive annotated events. Then, in the court view generator, we incorporated these events into the court view generation by merging the case fact and event as the new input. Besides, to alleviate the computational burden in EGG during inference that employs LLMs, we further proposed a LLMs-free EGG method based on the contrastive constraint. This enhancement enables court view generation without requiring event information during the inference phase. Experimental results on a real-world dataset clearly demonstrated the effectiveness of our proposed method.

Acknowledgements. This research was supported by grants from the National Natural Science Foundation of China (Grants No. 62337001, 623B1020) and the Fundamental Research Funds for the Central Universities.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *Proceedings of International Conference on Learning Representations (ICLR)* (2014).

[2] BaiChuan-Inc. 2023. A large-scale 7B pretraining language model developed by BaiChuan-Inc. https://github.com/baichuan-inc/Baichuan-7B.

[3] Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yiqun Liu, Yixing Fan, and Xueqi Cheng. 2023. A Unified Generative Retriever for Knowledge-Intensive Language Tasks via Prompt Learning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1448–1457.

[4] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on EMNLP*.

[5] Jiaxi Cui, Zongjian Li, Yang Yan, Bohua Chen, and Li Yuan. 2023. ChatLaw: Open-Source Legal Large Language Model with Integrated External Knowledge Bases. *ArXiv* (2023).

[6] Xingyi Duan, Baoxin Wang, Ziyue Wang, Wentao Ma, Yiming Cui, Dayong Wu, Shijin Wang, Ting Liu, Tianxiang Huo, Zhen Hu, et al. 2019. Cjrc: A reliable human-annotated benchmark dataset for chinese judicial reading comprehension. In *China National Conference on Chinese Computational Linguistics*. Springer, 439–451.

[7] Yi Feng, Chuanyi Li, and Vincent Ng. 2022. Legal Judgment Prediction via Event Extraction with Constraints. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

[8] Wanwei He, Jiabao Wen, Lei Zhang, Hao Cheng, Bowen Qin, Yunshui Li, Feng Jiang, Junying Chen, Benyou Wang, and Min Yang. 2023. HanFei-1.0. https://github.com/siat-nlp/HanFei.

[9] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

[10] Quzhe Huang, Mingxu Tao, Zhenwei An, Chen Zhang, Cong Jiang, Zhibin Chen, Zirui Wu, and Yansong Feng. 2023. Lawyer LLaMA Technical Report. *ArXiv* (2023).

[11] Weijing Huang, Xianfeng Liao, Zhiqiang Xie, Jiang Qian, Bojin Zhuang, Shaojun Wang, and Jing Xiao. 2020. Generating Reasonable Legal Text through the Combination of Language Modeling and Question Answering. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*.

[12] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, Vol. 1. 2.

[13] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

[14] Chuanyi Li, Yu Sheng, Jidong Ge, and Bin Luo. 2019. Apply Event Extraction Techniques to the Judicial Field. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*.

[15] 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. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 131–140.

[16] Qingquan Li, Qifan Zhang, Junjie Yao, and Yingjie Zhang. 2020. Event Extraction for Criminal Legal Text. In *2020 IEEE International Conference on Knowledge Graph (ICKG)*.

[17] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics.

[18] Yifei Liu, Yiquan Wu, Yating Zhang, Changlong Sun, Weiming Lu, Fei Wu, and Kun Kuang. 2023. ML-LJP: Multi-Law Aware Legal Judgment Prediction. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1023–1034.

[19] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

[20] Bingfeng Luo, Yansong Feng, Jianbo Xu, Xiang Zhang, and Dongyan Zhao. 2017. Learning to Predict Charges for Criminal Cases with Legal Basis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2727–2736.

[21] Yixiao Ma, Yunqiu Shao, Yueyue Wu, Yiqun Liu, Ruizhe Zhang, Min Zhang, and Shaoping Ma. 2021. LeCaRD: a legal case retrieval dataset for Chinese law system. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[22] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[23] OpenAI. 2023. Introducing ChatGPT. *OpenAI Blogs* (2023). https://openai.com/blog/chatgpt

[24] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 311–318.

[25] Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

[26] Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. (ACL)*.

[27] Yunqiu Shao, Yueyue Wu, Yiqun Liu, Jiaxin Mao, Min Zhang, and Shaoping Ma. 2021. Investigating user behavior in legal case retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[28] Shirong Shen, Guilin Qi, Zhen Li, Sheng Bi, and Lusheng Wang. 2020. Hierarchical Chinese Legal event extraction via Pedal Attention Mechanism. In *Proceedings of the 28th International Conference on Computational Linguistics*.

[29] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca.

[30] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971* (2023).

[31] 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* (2017).

[32] Yunhe Wang, Hanting Chen, Yehui Tang, Tianyu Guo, Kai Han, Ying Nie, Xutao Wang, Hailin Hu, Zheyuan Bai, Yun Wang, et al. 2023. PanGu: Enhancing Language Model Architectures via Nonlinearity Compensation. *arXiv preprint arXiv:2312.17276* (2023).

[33] Yiquan Wu, Kun Kuang, Yating Zhang, Xiaozhong Liu, Changlong Sun, Jun Xiao, Yueting Zhuang, Luo Si, and Fei Wu. 2020. De-Biased Court's View Generation with Causality. In *Proceedings of the 2020 Conference on EMNLP*.

[34] Yiquan Wu, Yifei Liu, Weiming Lu, Yating Zhang, Jun Feng, Changlong Sun, Fei Wu, and Kun Kuang. 2022. Towards Interactivity and Interpretability: A Rationale-based Legal Judgment Prediction Framework. (2022), 4787–4799.

[35] Feng Yao, Chaojun Xiao, Xiaozhi Wang, Zhiyuan Liu, Lei Hou, Cunchao Tu, Juanzi Li, Yun Liu, Weixing Shen, and Maosong Sun. 2022. LEVEN: A Large-Scale Chinese Legal Event Detection Dataset. In *Findings of the Association for Computational Linguistics: ACL 2022*.

[36] Hai Ye, Xin Jiang, Zhunchen Luo, and Wenhan Chao. 2018. Interpretable charge predictions for criminal cases: Learning to generate court views from fact descriptions. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics(NAACL)* (2018).

[37] Linan Yue, Qi Liu, Yichao Du, Weibo Gao, Ye Liu, and Fangzhou Yao. 2023. Fedjudge: Federated legal large language model. *arXiv preprint arXiv:2309.08173* (2023).

[38] Linan Yue, Qi Liu, Binbin Jin, Han Wu, and Yanqing An. 2024. A Circumstance-aware Neural Framework for Explainable Legal Judgment Prediction. *IEEE Transactions on Knowledge and Data Engineering* (2024).

[39] Linan Yue, Qi Liu, Binbin Jin, Han Wu, Kai Zhang, Yanqing An, Mingyue Cheng, Biao Yin, and Dayong Wu. 2021. NeurJudge: A Circumstance-aware Neural Framework for Legal Judgment Prediction. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[40] Linan Yue, Qi Liu, Han Wu, Yanqing An, Li Wang, Senchao Yuan, and Dayong Wu. 2021. Circumstances enhanced criminal court view generation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1855–1859.

[41] Shengbin Yue, Wei Chen, Siyuan Wang, Bingxuan Li, Chenchen Shen, Shujun Liu, Yuxuan Zhou, Yao Xiao, Song Yun, Wei Lin, et al. 2023. Disc-lawllm: Fine-tuning large language models for intelligent legal services. *arXiv preprint arXiv:2309.11325* (2023).

[42] Han Zhang, Zhicheng Dou, Yutao Zhu, and Ji-Rong Wen. 2023. Contrastive Learning for Legal Judgment Prediction. *ACM Transactions on Information Systems* 41, 4 (2023), 1–25.

[43] Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. 2018. Legal Judgment Prediction via Topological Learning. In *Proceedings of the 2018 Conference on EMNLP*.

[44] Haoxi Zhong, Yuzhong Wang, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. Iteratively questioning and answering for interpretable legal judgment prediction. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, Vol. 34. 1250–1257.