Unleashing the Power of LLMs in Court View Generation by Stimulating Internal Knowledge and Incorporating External Knowledge

Yifei Liu[1], **Yiquan Wu**[2], **Ang Li**[2], **Yating Zhang**[3], **Changlong Sun**[3], **Weiming Lu**[2], **Fei Wu**[2], **Kun Kuang**[2]
[1]School of Software Technology, Zhejiang University
[2]College of Computer Science and Technology, Zhejiang University
[3]Alibaba Group
{liuyifei,wuyiquan,leeyon,luwm,wufei,kunkuang}@zju.edu.cn,
yatingz89@gmail.com, changlong.scl@taobao.com

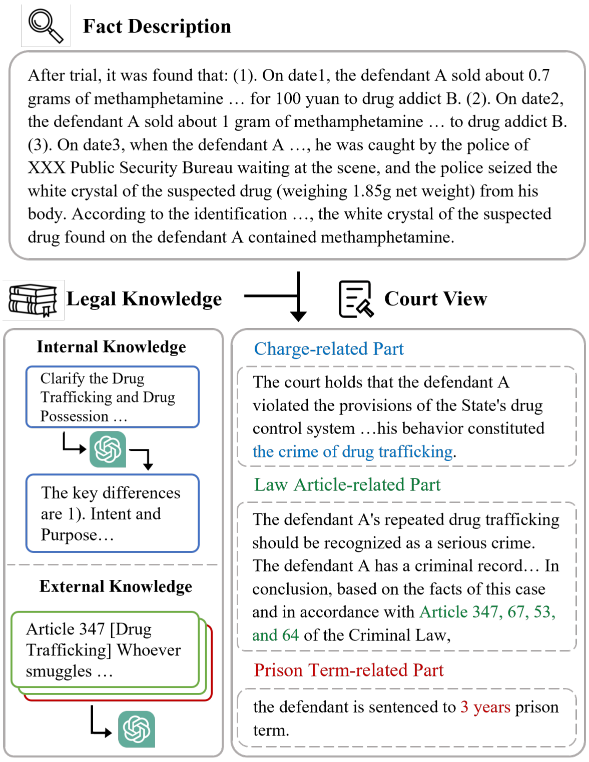
Abstract

Court View Generation (CVG) plays a vital role in the realm of legal artificial intelligence, which aims to support judges in crafting legal judgment documents. The court view consists of three essential judgment parts: the charge-related, law article-related, and prison term-related parts, each requiring specialized legal knowledge, rendering CVG a challenging task. Although Large Language Models (LLMs) have made remarkable strides in language generation, they encounter difficulties in the knowledge-intensive legal domain. Actually, there can be two types of knowledge: internal knowledge stored within LLMs' parameters and external knowledge sourced from legal documents outside the models. In this paper, we decompose court views into different parts, stimulate internal knowledge, and incorporate external information to unleash the power of LLMs in the CVG task. To validate our method, we conduct a series of experiment results on two real-world datasets LAIC2021 and CJO2022. The experiments demonstrate that our method is capable of generating more accurate and reliable court views.

1 Introduction

In Legal Artificial Intelligence (Legal AI), the task of Court View Generation (CVG) has been studied for years (Ye et al., 2018; Li and Zhang, 2021; Yue et al., 2021b), aiming to generate the judgment document based on the fact description of a legal case to assist judges in writing legal documents. The court view is mainly composed of three parts: the charge-related part, the law article-related part, and the prison term of the defendant. Each part is formed by the judgment result and rationale, as shown in the Fig. 1. Also, each part exhibits distinctive legal characteristics, requiring different legal knowledge, which makes CVG a challenging task.

The previous CVG works mostly focus on generating more fluent court views while ignoring the performance of the three judgment parts (Ye et al., 2018; Li and Zhang, 2021; Yue et al., 2021b). Also, the previous works require a large amount of data for training, while certain types of legal cases may face the problem of lacking data (e.g., there are only a few cases of large-scale corruption and bribery every year). As research on Large Language Models (LLMs) has progressed, the most recent iterations of LLMs have exhibited remarkable language generation capabilities. This drives us to utilize the strong generative capabilities of LLMs for the task of CVG, relying on just a few in-context examples.



After trial, it was found that: (1). On date1, the defendant A sold about 0.7 grams of methamphetamine … for 100 yuan to drug addict B. (2). On date2, the defendant A sold about 1 gram of methamphetamine … to drug addict B. (3). On date3, when the defendant A …, he was caught by the police of XXX Public Security Bureau waiting at the scene, and the police seized the white crystal of the suspected drug (weighing 1.85g net weight) from his body. According to the identification …, the white crystal of the suspected drug found on the defendant A contained methamphetamine.

Clarify the Drug Trafficking and Drug Possession …

The key differences are 1). Intent and Purpose…

Article 347 [Drug Trafficking] Whoever smuggles …

The court holds that the defendant A violated the provisions of the State's drug control system …his behavior constituted the crime of drug trafficking.

The defendant A's repeated drug trafficking should be recognized as a serious crime. The defendant A has a criminal record… In conclusion, based on the facts of this case and in accordance with Article 347, 67, 53, and 64 of the Criminal Law,

the defendant is sentenced to 3 years prison term.

Figure 1: An example case. The court view consists of three judgment parts. The text highlighted in color represents the judgment results, while the rest indicates the judgment rationale. The internal and external legal knowledge can be utilized by LLMs for enhanced generation.

However, the LLMs perform poorly in

Findings of the Association for Computational Linguistics: NAACL 2024, pages 2782–2792
June 16-21, 2024 ©2024 Association for Computational Linguistics

Task	Internal Knowledge	External Knowledge
Charge	Charge Clarifications	-
Law Article	-	Law Article Definitions
Prison Term	Code Generation Element Extraction	Penalty Principles

Table 1: Internal and external legal knowledge in the three judgment tasks.

knowledge-intensive legal domain tasks. How to effectively augment LLMs with legal knowledge bases to improve performance in the CVG task remains a challenge. In order to provide the judges more accurate and reliable court views, we unleash the power of LLMs by leveraging multiple legal knowledge bases. More specifically, we stimulate internal knowledge and incorporate external knowledge to construct multiple legal knowledge bases as shown in the Tab. 1, and then prompt the LLM to interact with these knowledge bases in the different judgment parts.

First, in the task of the charge-related part, the problem of confusing charges is a quite common issue in real judgment scenarios (Xu et al., 2020; Yue et al., 2021a). We consider that many lawyers have written numerous clarifications about confusing charges on the website, and we assume that this kind of knowledge has been crawled as part of the pre-training corpus of the LLMs. So we stimulate the internal knowledge of LLMs by prompting the LLMs to generate clarifications for all the charge pairs. Secondly, unlike charges, the label of law articles are meaningless index numbers (e.g. Article 347), which makes it difficult for LLMs to generate a sequence of numbers. To address this, given that the definitions of law articles are readily accessible, we collect these definitions as external knowledge. Subsequently, we retrieve the relevant law article definitions and append them to the input text, thereby incorporating external legal knowledge. Thirdly, in terms of the judgment of the prison term, many penalty principles have been released by the Supreme Court but often remain underutilized. Taking drug trafficking as an example, the prison term can be calculated precisely based on factors such as the amount of drugs, type of drugs, and so on. As a result, the prison term can be calculated in a symbolic way instead of learning a deep neural network with a large amount of data. Here we gather penalty principles corresponding to each charge as external knowledge. We utilize the internal ability of LLMs to generate Python code based on these penalty principles. In a given case, LLMs can extract relevant elements from the fact description and utilize them as parameters for calculating the prison term.

Overall, we decompose the CVG task into three distinct parts, employing distinct strategies for generation based on specific legal knowledge. Eventually, these three judgment parts are integrated to form a complete court view document.

Due to the fact that the pretraining corpus of LLMs primarily entails data up until 2021, we create a new dataset named CJO2022. This dataset is composed of criminal cases sourced from China Judgements Online in the year 2022. We conduct a series of experiments on the LAIC2021 dataset and the newly constructed CJO2022 dataset to validate our method. The experiments demonstrate that our method can generate more accurate and reliable court views and has competitive or even better performance against fully supervised state-of-the-art (SOTA) methods with only a few in-context examples.

In summary, our contributions can be outlined as follows:

- We firstly apply LLMs to the CVG task in the legal domain to enhance the generation of accurate and reliable court views.
- We decompose the CVG task into three parts, construct multiple legal knowledge bases, prompt LLMs to interact with them, aiming to stimulate internal and integrate external knowledge for enhanced generation.
- We construct a new dataset to avoid the potential problem of data leakage and conduct a series of experiments to validate the effectiveness of our method.

2 Related Work

2.1 Legal Artificial Intelligence

Legal Artificial Intelligence (Legal AI) aims to assist the legal professionals for the legal document work with artificial intelligence (Zhong et al., 2020). Early works focus on solving legal tasks from rule-based and symbol-based methods (Kort; Ulmer; Segal, 1984). Recently, Natural Language Processing (NLP) researchers concentrate more on data-driven and embedding methods and many NLP techniques have been applied to the legal domain for various legal tasks, such as Court View

Generation (Ye et al., 2018; Wu et al., 2020a; Yue et al., 2021b; Li and Zhang, 2021; Liu et al., 2023), Legal Judgment Prediction (Zhong et al., 2018; Yue et al., 2021a; Dong and Niu, 2021), Similar Case Matching (Peng et al., 2020; Yu et al., 2022b), Similar Case Retrieval (Ma et al., 2021, 2022; Li et al., 2023a) and Legal Event Extraction (Yao et al., 2022; Feng et al., 2022).

2.2 Court View Generation

Court View Generation (CVG) is an important task in Legal Artificial Intelligence. Given the fact description of a legal case, the target of CVG is to generate the court view, which is also the final judgment document about the case. In recent years, many research work has been devoted to this task. Ye et al. (2018) propose a label-conditioned sequence to sequence model for the court view generation. Wu et al. (2020a) use counterfactual decoders to generate judgment-discriminative court's views. Yue et al. (2021b) split the court view into adjudging circumstance and sentencing circumstance and uses two generators to generate the circumstances enhanced court views. Li and Zhang (2021) exploits the charge and law article information in the generation process and uses a Transformer-based architecture for generating the court view. All the above methods are based on deep neural networks, requiring a large amount of data for training, and they ignore the performance of the three judgment parts in the court view.

2.3 Large Language Models

Large Language Models (LLMs) have revolutionized the field of natural language processing recently (Ouyang et al., 2022; Du et al., 2022; Touvron et al., 2023), driving significant advancements in tasks such as machine translation, question-answering, text generation and more (OpenAI, 2023). While LLMs have demonstrated remarkable abilities in various NLP tasks, they still perform poorly when it comes to domain-specific tasks (e.g. legal or medical tasks) (Trautmann et al., 2022; Yu et al., 2022a). Some research works (Cui et al., 2023; Li et al., 2023b) continue fine-tuned on domain-specific corpus but requiring a large amount of high-quality data and high-cost GPU resources. In this paper, we aim to bridge the gap between LLMs and legal domain knowledge in the task of CVG. All the above LLMs can be enhanced by better utilization of internal and external legal knowledge.

3 Method

In this section, we decompose the Court View Generation (CVG) task into three parts and describe our method for each part. We stimulate the internal knowledge stored in the LLMs and incorporate external legal knowledge to generate the different judgment parts, as shown in the Tab. 1.

3.1 Problem Definition

For a legal case, we input the fact description f to generate the court view v, which consists of the three parts as shown in the Fig. 1: the charge-related part c, the law article-related part a and the prison term-related part p. For the charge-related part c, we define the c_{rs} as the judgment result and c_{rt} as the judgment rationale. Similarly, we define the a_{rs} and a_{rt} for the law article-related part a. the court view is decomposed into three parts and colored texts are judgment results and the rest are judgment rationales.

3.2 Retrieved Similar Cases

According to the principle of "treat like cases alike", we use a retriever to retrieve the similar cases as in-context examples. We sample 7,837 legal cases from the LAIC2021 dataset to create a relatively uniform distribution across labels, utilizing a dense retriever to encode the fact descriptions of these cases into embeddings for the similar case pool.

The retrieved cases serve as references for generating the court view, aligning the LLMs with the writing style of the court view. During inference, we first encode the fact description of the current case into an embedding, then calculate the cosine similarity between the embeddings of the current case and the similar cases, and finally select the top two cases as the in-context examples. Here, we use Contriever (Izacard et al., 2022) as the retriever.

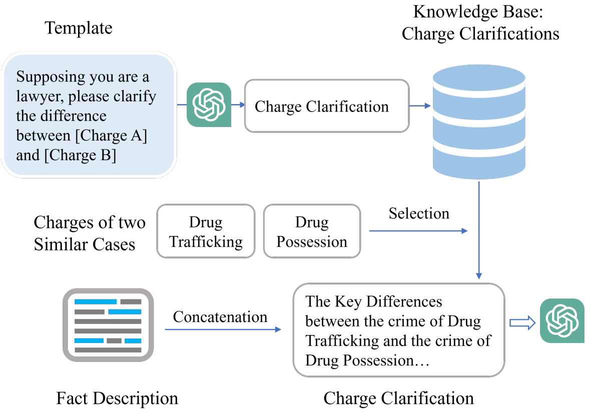

Template
Knowledge Base:
Charge Clarifications
Supposing you are a lawyer, please clarify the difference between [Charge A] and [Charge B]
Charge Clarification
Charges of two Similar Cases
Drug Trafficking
Drug Possession
Selection
Concatenation
The Key Differences between the crime of Drug Trafficking and the crime of Drug Possession…
Fact Description
Charge Clarification

Figure 2: Charge-related Part Generation.

The Key Differences between the crime of Drug Trafficking and the crime of Drug Possession:	
1. Intent and Purpose	Drug possession is typically for personal use, while drug trafficking involves ...
2. Quantities	Drug trafficking often involves larger quantities …
3. Legal Consequences	Drug trafficking generally carries more severe legal penalties ...
4. Aggravating Factors	Drug trafficking might include involvement in organized crime ...

Figure 3: Charge Clarification from *gpt-3.5-turbo*.

3.3 Charge-related Part Generation

For the charge-related part generation, we stimulate the internal knowledge stored in the LLMs. The problem of confusing charge have been studied for years (Xu et al., 2020; Yue et al., 2021a). However, there have been many clarifications of confusing charges written by lawyers (or other legal experts), which can be easily crawled as part of the corpus and used for pretraining the LLMs. So, we listed all the charge pairs and directly prompted the LLMs to generate clarification for each charge pair.

For example, given the charge pair "the crime of Drug Trafficking" and "the crime of Drug Possession," using the template as illustrated in the Fig. 2, we prompted the LLM to generate clarifications for the two. The response from the LLM is shown in the Fig. 3, the LLM lists the difference between the two charges from multiple aspects. After verification, the charge clarifications generated by the LLM are of high quality. We perform the above operations as a preliminary work to stimulate the internal knowledge of LLMs and construct the knowledge base of charge clarifications.

Given a case, we select the clarification of the charges of the retrieved two similar cases from the charge clarification knowledge base. Then, we concatenate the two similar cases, clarifications of the two charges, and the fact description of the current case as input for the LLMs to generate the judgment result and rationale for the charge.

3.4 Law Article-related Part Generation

In terms of the law article-related part, we utilize law article definitions as external knowledge for the generation. Unlike the judgment results of charges, which already contain semantic information (e.g., drug trafficking), the labels of law articles are index numbers in the Code of Law that contain no information (e.g., Article 347). Therefore, it's difficult for LLMs to generate a sequence of meaningless index numbers without knowing the specified definitions behind these indexes. As a result, we gather all the law article definitions as external legal knowledge and obtain the definition of the relevant law articles as part of the context when prompting, as illustrated in the Fig. 4.

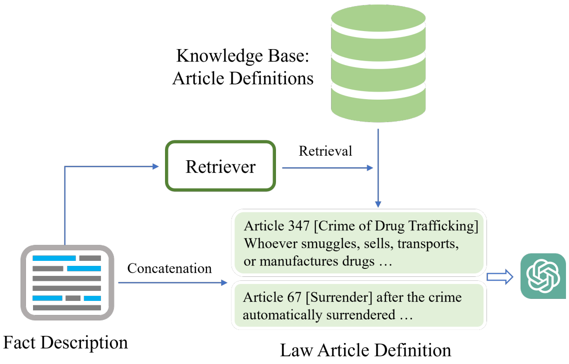

Knowledge Base:
Article Definitions
Retriever
Retrieval
Article 347 [Crime of Drug Trafficking] Whoever smuggles, sells, transports, or manufactures drugs …
Concatenation
Article 67 [Surrender] after the crime automatically surrendered …
Fact Description
Law Article Definition

Figure 4: Law Article-related Part Generation.

There are two parts of relevant articles: 1) law articles cited in the similar cases, and 2) retrieved law articles from a retriever. For the first part, we obtain all the indexes of law articles cited in the similar cases. For the second part, we use a dense retriever to retrieve the most relevant law articles based on the cosine similarity between the fact description and the law article definitions. We merge the two parts and select four law articles as the relevant law articles. The retriever we use here is also Contriever (Izacard et al., 2022).

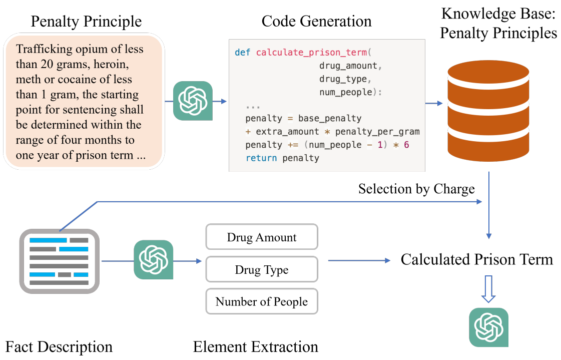
When prompting, we concatenate the two similar cases, definitions of relevant law articles, and the fact description of the current case as input to generate law articles and rationale for citing these law articles.

3.5 Prison Term Calculation

In the part of the prison term, we utilize both internal and external knowledge to enhance the LLMs. According to previous work, the performance of prison term prediction has been very unsatisfactory. The main reason could be attributed to the lack of external knowledge (e.g., the detailed penalty principles released by the Supreme People's Court).

Also, previous works mostly use black-box neural networks to learn the penalty judgment process in a data-driven way, which makes it cumbersome to learn the prison term predictor. In reality, the prison term can be calculated more precisely based on the penalty principles in a symbolic way without requiring large amounts of data.

Take the crime of drug trafficking as an example, when the drug amount surpasses a specified limit, a



Trafficking opium of less than 20 grams, heroin, meth or cocaine of less than 1 gram, the starting point for sentencing shall be determined within the range of four months to one year of prison term ...

```
def calculate_prison_term(
                drug_amount,
                drug_type,
                num_people):
  ...
  penalty = base_penalty
  + extra_amount * penalty_per_gram
  penalty += (num_people - 1) * 6
  return penalty
```

Figure 5: Prison Term Calculation.

base penalty is imposed. Subsequently, the portion of the drug exceeding this limit is used for calculating an additional penalty based on specific rules (e.g., three months for every extra gram).

There are numerous detailed penalty principles that vary among different charges. As a result, we take the external knowledge, document of penalty principles, as the input and utilize the internal code generation capability of LLMs to produce Python code prison term calculations. Subsequently, the generated codes are verified by a small group of human experts and collected as the knowledge base of prison term.

The inference process is illustrated in the Fig. 5. Given judgment results for charge, we select the corresponding Python code from the knowledge base. Subsequently, we use the pre-defined questions associated with this charge to instruct the LLMs in extracting pertinent elements from the fact description, utilizing them as parameters for the Python code used in calculating the prison term. To ensure the accuracy of element extraction, we generate it 10 times and vote for the majority as the final answer for each element (Wang et al., 2023).

Because the prison term-related part is relatively fixed, we use a template with the inserted prison term as the final part of court view (e.g. *the defendant is sentenced to fixed-term imprisonment of* {calculated prison term}).

3.6 Merged Court View

Finally, we merge the three parts from above into a complete court view. Notably, we decompose the process of court view generation into multiple sub-processes. All the intermediate generation results can be interacted with and modified by human judges to avoid further error propagation in the following steps.

4 Experiments

4.1 Datasets

We randomly sampled 3,936 cases from LAIC2021 dataset as testset which consists of fact description, court view and we use the rest data for the training of baseline models. Additionally, to mitigate the potential issue of data leakage in the pre-training stage of LLMs, where the corpus was collected before 2021, we newly crawled 2,122 cases from Chinese Judgment Online in 2022 as another testset. The detailed statistics of the two datasets are shown in the Tab. 3.

4.2 Baselines

4.2.1 Fully-Supervised Mehods

We implement the following fully-supervised methods as baselines. As general generation method, **BART** (Lewis et al., 2020) is trained by corrupting text with an arbitrary noising function, and learning a model to reconstruct the original text. For the CVG method, **C3VG** (Yue et al., 2021b) split the court view into adjudging circumstance and sentencing circumstance and uses two generators to generate the circumstances enhanced court view. To better evaluate the the generated court view, we take the accuracy of the three judgment results into consideration and implement the following predictive methods for comparison. **Bert** (Devlin et al., 2019) , **Roberta** (Liu et al., 2019) and **Electra** (Clark et al., 2020), are all masked language models used for natural language understanding tasks. Especially, we use a legal version of Electra which continue pretrained on a legal corpus. **ML-LJP** (Liu et al., 2023) extracts the label-specific features of the fact and applies a graph attention network to capture the high-order interactions among multiple law articles. The amount of data for training fully-supervised methods is 79,169.

4.2.2 Large Language Models

For the LLMs we use **Dav002**, **Dav003** and **GPT3.5** which are all LLMs API provided by OpenAI and refers to *text-davinci-002*, *text-davinci-003* and *gpt-3.5-turbo* respectively.

For the ablation settings, **0-shot** refers to directly prompting LLMs to generate court view with no in-context examples; **2-shot** refers to generating court view with two retrieved in-context examples; **2-shot w/ kb** refers to prompting LLMs to generate court view with two shots and enhanced by knowledge bases in different part of generation.

Method	Shot	LAIC2021						CJO2022					
		Charge		Law Article		Prison Term		Charge		Law Article		Prison Term	
		MiF	MaF	MiF	MaF	MaF	Acc25	MiF	MaF	MiF	MaF	MaF	Acc25
BERT	Full-shot	95.57	89.19	92.11	57.56	32.14	38.26	91.36	83.12	74.26	58.68	30.22	33.34
Roberta	Full-shot	**96.34**	90.94	92.91	59.69	35.09	42.37	**92.48**	81.63	77.92	61.90	33.63	35.10
Electra(Legal)	Full-shot	96.15	89.34	89.74	57.28	33.74	38.12	91.52	83.68	75.14	61.97	35.71	35.45
ML-LJP	Full-shot	96.06	90.96	**93.15**	60.10	36.52	-	92.29	83.10	**78.90**	**62.60**	35.91	-
Dav002	0-shot	54.83	47.68	0.37	0.22	5.53	5.20	40.62	31.01	0.19	0.01	1.25	2.95
	2-shot	82.34	81.47	50.48	37.94	13.55	13.82	69.14	56.85	31.74	19.63	7.43	8.99
	2-shot w/ kb	85.08	82.22	61.04	56.33	32.11	38.17	72.27	56.08	54.89	44.62	24.22	29.74
Dav003	0-shot	67.22	62.23	0.61	0.42	5.65	4.92	46.51	35.28	0.43	0.17	4.33	3.86
	2-shot	89.29	87.69	65.99	48.69	18.47	19.35	78.75	68.07	51.88	33.12	17.26	19.13
	2-shot w/ kb	91.53	90.18	76.60	64.07	39.82	45.60	81.64	79.92	69.88	57.02	36.34	35.74
GPT3.5	0-shot	73.66	64.86	7.51	2.77	10.90	15.98	80.96	59.37	9.47	1.95	11.00	9.44
	2-shot	93.24	92.55	72.92	58.24	19.16	24.31	83.49	71.80	57.08	26.39	24.03	26.07
	2-shot w/ kb	93.73	**93.12**	83.31	**65.18**	**44.12**	**49.26**	90.13	**88.39**	74.13	62.27	**39.34**	**43.74**

Table 2: Results of three judgment results on the two datasets, the best is **bolded** and the second best is underlined.

4.3 Experiments Settings

All the baseline models are trained with the settings in their original paper on a server with 4x3090 GPUs. For the settings of LLMs, we set *top_p* and *temperature* to the default 1.

For the evaluation of charge results , we employ Micro F1 score (MiF) and Macro F1 score (MaF) in single-label classification. For law articles results, we use Micro F1 score (MiF) and Macro F1 score (MaF) in multi-label classification for evaluation. For evaluating the results of prison term, since we calculate prison term accurate to the month, we adopt regression metrics Acc25 for evaluation. Acc25 refers to predicted value will be considered as correct if it is within the upper and lower 25% range of the correct value which is calculated as Acc25= $\frac{|\hat{y}-y|}{y} \leq 0.25$.

To ensure a fair comparison with previous work, we also convert the prison term in our method into non-overlapping intervals and evaluate it using the MaF metric. For the generation, we use ROUGE [1] (Lin, 2004) and BLEU [2] (Papineni et al.) as metrics for automatic evaluation.

Type	LAIC2021	CJO2022
# Test Samples	3,936	2,122
# Charge	50	44
# Law Article	69	70
Avg. tokens in Fact Description	338.6	265.1
Avg. tokens in Court View	177.5	203.8

Table 3: Dataset Statistics.

4.4 Experiments Results

4.4.1 Comparison against Baselines

The evaluation of judgment results are presented in the Tab. 2. We can draw the following conclusions: 1) For the results of charge and law articles, baseline models with full training set still performs well, as these two tasks are relatively straightforward for fully fine-tuned models. Despite this, our LLM-based method demonstrates competitive performance with only two shots, and performs better on MaF, suggesting that our approach excels in predicting low-frequency labels. 2) In the context of prison term prediction, the baseline models struggle to predict the correct prison term when trained directly in a data-driven way without external penalty principles provided. Our method fully utilizes penalty principles to calculate the prison term in a symbolic way, avoiding the need for a large amount of data for training and achieving better performance. Therefore, our approach enables the generation of accurate and reliable court views.

The generation results are shown in the Tab. 4. Compared to the fully-supervised baseline models, our method demonstrates improved performance across most metrics with only two in-context examples. The baseline methods perform better on R1 and B1 scores, indicating that traditional methods tend to overfit on frequent tokens, whereas our method excels in RL and BN scores, showcasing improved performance in generating n-grams and reasonable sentences, benefiting from the powerful language generation capabilities of LLMs. Consequently, our approach is capable of generating fluent and comprehensible court views as well.

[1] https://pypi.org/project/rouge/

[2] https://www.nltk.org/api/nltk.translate.bleu_score.html

Method	Shot	LAIC2021						CJO2022					
		ROUGE			BLEU			ROUGE			BLEU		
		R1	R2	RL	B1	B2	BN	R1	R2	RL	B1	B2	BN
BART	Full-shot	58.67	45.62	55.49	60.85	53.97	49.03	54.48	42.62	41.23	57.38	50.01	45.30
C3VG	Full-shot	**67.52**	48.85	65.60	**67.52**	58.17	55.77	61.19	45.02	55.34	**63.93**	53.66	51.23
Dav002	0-shot	18.71	5.09	11.89	23.81	14.75	11.11	13.67	8.16	10.45	20.65	11.71	8.51
	2-shot	48.62	26.84	46.36	46.49	36.52	32.38	44.06	32.76	43.34	36.01	30.24	26.58
	2-shot w/ kb	49.00	27.43	47.72	46.77	37.00	33.17	43.18	36.41	44.29	38.49	31.95	25.49
Dav003	0-shot	25.64	9.68	17.95	21.73	14.50	11.92	18.19	6.99	12.12	11.83	6.23	5.31
	2-shot	57.84	41.98	56.81	54.69	47.01	44.34	51.63	42.28	48.78	50.50	46.75	45.02
	2-shot w/ kb	64.45	**50.88**	64.39	59.91	53.68	51.47	50.36	43.50	52.37	53.53	48.55	46.55
GPT3.5	0-shot	40.43	20.45	30.86	38.95	22.23	20.98	29.86	13.33	21.37	26.64	19.11	16.33
	2-shot	61.53	46.59	60.16	60.24	58.18	52.55	62.06	**52.01**	**60.21**	62.06	56.18	55.74
	2-shot w/ kb	67.46	50.84	**66.56**	64.82	**61.89**	**57.42**	**62.89**	51.46	58.14	62.09	**56.55**	**57.30**

Table 4: Court View Generation results on the two datasets, the best is **bolded** and the second best is underlined.

4.4.2 Ablation Study

We also conduct a series of ablation experiments as shown in the Tab. 2 and Tab. 4. For the charge-related part, the clarification of charges aids LLMs in better distinguishing confusing charges and improves performance. For the law article-related part, LLMs have no knowledge of how to predict the correct law articles in **0-shot** setting. The reason is that, for charges, LLMs can reason from the names of charge labels, but the names of law articles are meaningless index numbers. In **2-shot** setting, although the performance are boosted, LLMs still don't know the meaning of index numbers, but simply copy the numbers from the in-context examples. The result of **2-shot w/ kb** indicates that the retrieved definitions of law articles from the external knowledge base assist LLMs in understanding the meaning behind the index numbers. For the prison term-related part, utilizing extracted elements and external penalty principles, we transform the judgment process of prison terms into information extraction and code generation problems where LLMs excel, calculating the prison term symbolically. This approach significantly improves the performance of LLMs on the prison term task.

Charge	Method	Shot	Classification	Regression
			MaF	Acc25
Fraud	ML-LJP	Full-shot	4.166	-
	GPT3.5	2-shot	4.76	18.30
		2-shot w/ kb	**24.82**	**39.01**
Drug Trafficking	ML-LJP	Full-shot	12.96	-
	GPT3.5	2-shot	8.33	11.21
		2-shot w/ kb	**53.33**	**71.05**

Table 5: Results of prison term calculation on two specified charges, the best is **bolded**.

4.4.3 Prison Term Calculation Study

To assess the performance of our prison term calculation method on different charges, we specifically choose two charges (Fraud and Drug Trafficking) to showcase the results. As shown in the Tab. 5, our method substantially outperforms the fully-supervised method ML-LJP and large language model GPT3.5. It can also be observed that our method performs better on Drug Trafficking than on Fraud. The reason is that, in the case of Fraud, the primary element for calculating the prison term is the amount of fraud committed by the defendant, and accurately extracting this information during the element extraction stage is difficult, subsequently affecting the calculation of prison term.

For example, some amount numbers appear repeatedly, some values of the fraud items are not explicitly labeled, or the defendant returned a portion of the victim's money after his arrest. But for the crime of Drug Trafficking, the amount of drugs can be extracted more accurately. As a result, the element extraction in legal cases varies in difficulty for different charges and affects the performance of downstream tasks, which can be further studied in future work.

4.5 Human Evaluation

To further study the performance of the generation results, we also randomly sample 200 cases from each dataset for human evaluation. We adopt

Fact Description		The trial found that: (1). One night at … the defendant A, through prior telephone contact, rushed to a room in the Hotel. The two small tubes of about 1.04 grams of methamphetamine were sold to drug users B at a price of 2,000 yuan. (2). One night at … defendant A sold two small tubes of about 1.04 grams of methamphetamine to drug users B at a price of 2,000 yuan. (3). In the early morning of … defendant A sold 2 small tubes of methamphetamine to drug users B at a price of 2000 yuan, B was caught by the police on the spot … After identification, the above drug suspects and pills contain methamphetamine components, … a total of 12.70 grams. In summary, the defendant A sold drugs three times, with a total weight of about 14.78 grams.
Court View	Ground-Truth	The Court held that the defendant A, knowing that it was the drug methamphetamine, still illegally sold it on several occasions, and the amount reached more than 10 grams, and his conduct constituted the crime of drug trafficking. The charges charged by the public prosecution are upheld. The defendant A was able to truthfully confess his crimes after being brought to trial, and was given a lighter punishment according to law … In accordance with Articles 347, 67 and 64 of the Criminal Law of the People's Republic of China, the defendant was sentenced to fixed-term imprisonment of 7 years and 6 months.
	C3VG	The court believes that the defendant A sold 2 small tubes of methamphetamine to a drug addict at a price of 2,000 yuan for the purpose of illegal possession, and his behavior has constituted the crime of drug trafficking ... According to the law, punishment may be mild. In accordance with the provisions of Article 267, 62 and 73 of the Criminal Law of the People's Republic of China, he is sentenced to fixed-term imprisonment for three years.
	Ours	The Court held that the defendant A repeatedly sold the drug methamphetamine totaling about 14.78 grams, … constituting the crime of drug trafficking. The public prosecution accused the defendant A of drug trafficking and was found guilty. The defendant A truthfully confessed his crime after the case … Accordingly, in accordance with the provisions of Articles 347, 67 and 64 of the Criminal Law of the People's Republic of China, the defendant was sentenced to fixed-term imprisonment of seven years and six months …

Figure 6: Case Study. The blue indicates the judgment result is correct and the red indicates wrong.

Consistency and **Fluency** as the metrics.

Consistency measures the consistency between the judgment rationale and judgment result and **Fluency** measures the fluency of the generated court view. Three annotators are asked to give scores for the two metrics, where 1 denotes the lowest score and 5 denotes the best score. As shown in the Tab. 6, the LLM-based methods demonstrate better performance in terms of both metrics, and our method further enhances the effectiveness of the LLMs.

Method	Shot	LAIC2021		CJO2022	
		Cons.	Flue.	Cons.	Flue.
C3VG	Full-shot	4.12	4.40	3.89	4.27
GPT3.5	2-shot	4.22	4.81	4.10	4.75
	2-shot w/ kb	**4.51**	**4.89**	**4.39**	**4.81**

Table 6: Results of human evaluations.

4.6 Case Study

Here we use a case to compare our method with the baseline model as shown in the Fig. 6. The baseline method, C3VG, can generate fluent and comprehensive court views. However, a notable issue is that the outcomes of the three judgment parts are often incorrect, thus, it cannot form an accurate and reliable court view. Our approach addresses this by leveraging the capabilities of LLMs to generate well-structured court views while enhancing the accuracy of the three judgment parts by stimulating the internal legal knowledge and incorporating the external legal knowledge.

5 Conclusion and Future Work

In this paper, we explore the Court View Generation (CVG) task, leveraging the powerful generation capabilities of LLMs. To apply LLMs in the knowledge-intensive CVG task, we decompose the CVG into three different parts and construct multiple legal knowledge bases by stimulating internal knowledge and incorporating external knowledge. The LLMs are enhanced by interacting with different legal knowledge bases in different sub-processes to generate more accurate and reliable court views. The experiments on two real-world datasets validate the effectiveness of our method. In the future, we will explore applying LLMs to other legal tasks (e.g., Legal Elements Extraction, etc.) and combining LLMs with a broader range of legal knowledge to adapt to different legal tasks.

6 Ethical Issue Discussion

With the development of LegalAI, ethical issues become more important since any subtle miscalculation may trigger serious consequences (Wu et al., 2020b). The target user of CVG is the trial judge, who suffers from a 'daunting workload'. In such circumstances, the proposed method aims to offer suggestions to the judges but should **never replace** the human judges. Since our method divides the entire generation task into multiple sub-processes, human judges can interact with the intermediate results and correct possible errors in the intermediate process to avoid further error propagation.

7 Limitations

Retriever	Hit@1	Hit@2	Hit@3	Hit@4	Hit@5
Contriever	**83.90**	**88.04**	**91.46**	**93.41**	**94.14**
SimCSE	74.63	82.20	85.37	87.80	89.76
BM25	52.92	61.21	63.65	65.60	67.31

Table 7: Hit@K of different retrievers in retrieving cases with the same charge as the current case.

Due to the context limitation of LLMs and the lengthy nature of legal cases, only two similar cases are included in the prompt. According to the Tab. 7, incorporating more similar cases can lead to better performance. Leveraging the long-context techniques of LLMs (Ding et al., 2023; Xu et al., 2023; Chen et al., 2023), we can include more similar cases in the prompt. In addition, we present the performance of different unsupervised universal retrievers in the Tab. 7. A more adept retriever tailored to the legal domain can further enhance the performance of downstream tasks. We will leave it as future work.

References

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. 2023. Longlora: Efficient fine-tuning of long-context large language models. *CoRR*, abs/2309.12307.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Jiayu Ding, Shuming Ma, Li Dong, Xingxing Zhang, Shaohan Huang, Wenhui Wang, Nanning Zheng, and Furu Wei. 2023. Longnet: Scaling transformers to 1, 000, 000, 000 tokens. *CoRR*, abs/2307.02486.

Qian Dong and Shuzi Niu. 2021. Legal judgment prediction via relational learning. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 983–992. ACM.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. GLM: general language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 320–335. Association for Computational Linguistics.

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), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 648–664. Association for Computational Linguistics.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022.

Fred Kort. Predicting supreme court decisions mathematically: A quantitative analysis of the "right to counsel" cases. *American Political Science Review*, 51(1):1–12.

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, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Haitao Li, Qingyao Ai, Jia Chen, Qian Dong, Yueyue Wu, Yiqun Liu, Chong Chen, and Qi Tian. 2023a. SAILER: structure-aware pre-trained language model for legal case retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 1035–1044. ACM.

Quanzhi Li and Qiong Zhang. 2021. Court opinion generation from case fact description with legal basis. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14840–14848. AAAI Press.

Yunxiang Li, Zihan Li, Kai Zhang, Ruilong Dan, and You Zhang. 2023b. Chatdoctor: A medical chat model fine-tuned on llama model using medical domain knowledge. *CoRR*, abs/2303.14070.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Meeting of the Association for Computational Linguistics,Meeting of the Association for Computational Linguistics*.

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, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 1023–1034. ACM.

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. *CoRR*, abs/1907.11692.

Yixiao Ma, Qingyao Ai, Yueyue Wu, Yunqiu Shao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2022. Incorporating retrieval information into the truncation of ranking lists for better legal search. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 438–448. ACM.

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 *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 2342–2348. ACM.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation.

Dunlu Peng, Jiyin Yang, and Jing Lu. 2020. Similar case matching with explicit knowledge-enhanced text representation. *Appl. Soft Comput.*, 95:106514.

Jeffrey A. Segal. 1984. Predicting supreme court cases probabilistically: The search and seizure cases, 1962-1981. *American Political Science Review*, page 891–900.

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

Dietrich Trautmann, Alina Petrova, and Frank Schilder. 2022. Legal prompt engineering for multilingual legal judgement prediction. *CoRR*, abs/2212.02199.

S. Sidney Ulmer. Quantitative analysis of judicial processes: Some practical and theoretical applications. *Law and Contemporary Problems*, page 164.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

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

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

Nuo Xu, Pinghui Wang, Long Chen, Li Pan, Xiaoyan Wang, and Junzhou Zhao. 2020. Distinguish confusing law articles for legal judgment prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 3086–3095. Association for Computational Linguistics.

Peng Xu, Wei Ping, Xianchao Wu, Lawrence McAfee, Chen Zhu, Zihan Liu, Sandeep Subramanian, Evelina Bakhturina, Mohammad Shoeybi, and Bryan Catanzaro. 2023. Retrieval meets long context large language models. *CoRR*, abs/2310.03025.

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. *CoRR*, abs/2203.08556.

Hai Ye, Xin Jiang, Zhunchen Luo, and Wenhan Chao. 2018. Interpretable charge predictions for criminal cases: Learning to generate court views from fact descriptions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1854–1864. Association for Computational Linguistics.

Fangyi Yu, Lee Quartey, and Frank Schilder. 2022a. Legal prompting: Teaching a language model to think like a lawyer. *CoRR*, abs/2212.01326.

Weijie Yu, Zhongxiang Sun, Jun Xu, Zhenhua Dong, Xu Chen, Hongteng Xu, and Ji-Rong Wen. 2022b. Explainable legal case matching via inverse optimal transport-based rationale extraction. In *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 657–668. ACM.

Linan Yue, Qi Liu, Binbin Jin, Han Wu, Kai Zhang, Yanqing An, Mingyue Cheng, Biao Yin, and Dayong Wu. 2021a. Neurjudge: A circumstance-aware neural framework for legal judgment prediction. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 973–982. ACM.

Linan Yue, Qi Liu, Han Wu, Yanqing An, Li Wang, Senchao Yuan, and Dayong Wu. 2021b. Circumstances enhanced criminal court view generation. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 1855–1859. ACM.

Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. 2018. Legal judgment prediction via topological learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3540–3549. Association for Computational Linguistics.

Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. How does NLP benefit legal system: A summary of legal artificial intelligence. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5218–5230. Association for Computational Linguistics.