Fidelity-Constrained Decoding for Legal Tasks in Large Language Models without Finetuning

Anonymous ACL submission

Abstract

Large language models (LLMs) for legal tasks are designed to assist judges and lawyers in decision-making, where ensuring fidelity to case facts and legal elements is crucial for generating reliable legal interpretations and accurate predictions. However, existing methods, including prompt-based and fine-tuning approaches, either require extensive human effort or lack an explicit mechanism to enforce fidelity in model outputs. To address these challenges, we propose Fidelity-Constrained Decoding (FCD), a tuning-free framework that constrains the decoding process to maintain strict alignment with case facts and legal elements. Extensive experiments on three datasets using two open-domain LLMs show that FCD consistently enhances legal performance.

1 Introduction

Legal tasks such as legal judgment prediction (LJP) (Shui et al., 2023; Dong and Niu, 2021; Feng et al., 2022), legal document proofreading (LDP) (Liu and Luo, 2024; Kuleshov et al., 2020), and legal trigger words detection (LTD) (Yao et al., 2022; Fei et al., 2023), is a specialized research domain designed to assist practitioners in making legal decisions. In this context, fidelity—referring to a model's ability to generate outputs that are not only accurate but also faithful to the underlying case facts and legal elements (Ma et al., 2021; Yu et al., 2022)—is particularly important.

Recently, large language models (LLMs) have demonstrated impressive performance across various tasks (Chang et al., 2024; Bai et al., 2023), leading researchers to explore their adaptation to the legal domain (Blair-Stanek et al., 2023; Nay, 2023). One prominent approach is the prompt-based method (Jiang and Yang, 2023; Shui et al., 2023; Deng et al., 2023a), which involves creating carefully crafted legal prompts to guide LLMs in generating legally appropriate responses. While this method yields promising results in producing contextually relevant outputs, it requires extensive effort in prompt engineering and often struggles to ensure the fidelity of these outputs. As illustrated in Figure 1, for three common legal tasks LJP, LDP and LTD, the prompt-based method frequently generates free-form text that is often informal and lacks the fidelity needed for these tasks.

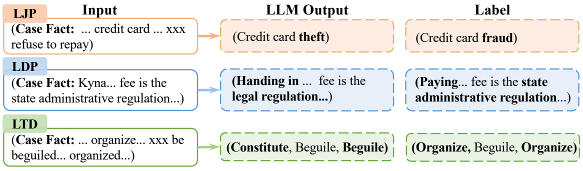
LJP
Input
LLM Output
Label
(Case Fact: ... credit card ... xxx refuse to repay)
(Credit card theft)
(Credit card fraud)
LDP
(Case Fact: Kyna... fee is the state administrative regulation...)
(Handing in ... fee is the legal regulation...)
(Paying... fee is the state administrative regulation...)
LTD
(Case Fact: ... organize... xxx be beguiled... organized...)
(Constitute, Beguile, Beguile)
(Organize, Beguile, Organize)

Figure 1: Examples of the unfaithful outputs from LLM. In **LJP**, LLM fabricates a non-existent charge. In **LDP**, LLM simplifies the legal terminology into colloquial language. In **LTD**, LLM fabricates trigger words not present in the input case and repeats words that only appear once.

The alternative method is fine-tuning, where large-scale legal data are employed to inject legal knowledge into LLMs (Yue et al., 2023; Wu et al., 2023a; Cui et al., 2023). This method implicitly encourages the models to adhere to legal facts and principles through next-token prediction. Despite its effectiveness, this approach requires extensive, high-quality data to adapt to various downstream legal tasks and lacks an explicit mechanism to guarantee the fidelity of LLMs' outputs.

In this paper, we propose a novel research perspective to address the challenges of fidelity and efficiency in legal tasks by focusing on constraining the decoding process of LLMs, rather than relying on traditional methods like prompt engineering or fine-tuning to inject legal knowledge. Our approach, called **F**idelity-**C**onstrained **D**ecoding (FCD), ensures that only a specific subset of tokens that align with case facts and legal elements is generated. This makes our method versatile and compatible with LLMs in a retrieval-augmented generation (RAG) manner, thereby enhancing their

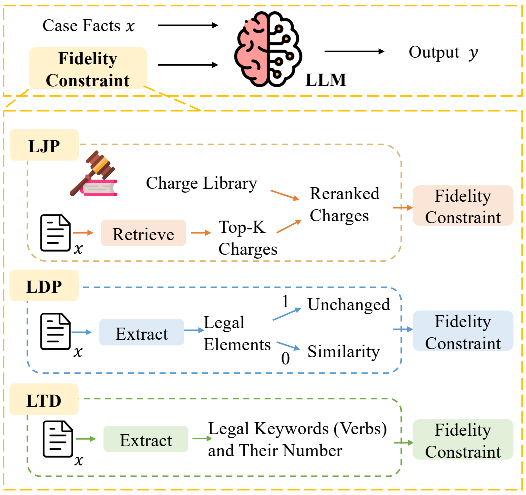

Figure 2: Overview of our FCD Framework. The LLM processes case facts x and applies the **Fidelity Constraint** specific to the legal task to generate output y.

performance across various legal tasks.

To validate the effectiveness of FCD, we conduct extensive experiments on three specific legal task datasets using two open-domain LLMs. The experimental results, evaluated under both zero-shot and RAG settings, demonstrate that FCD consistently enhances legal performance.

2 Method

2.1 Task Formulation

When employing LLMs for legal tasks, the LLMs receive prompts from specific tasks along with the legal case facts and generate outputs:

$$y = f_{\text{LLM}}(x, \text{prompt}; \theta), \tag{1}$$

where θ is the parameters of an open-domain LLM; prompt is the prompt related to the specific legal task; x is usually a legal case fact; y denotes the output results. Related work is in Appendix A.

2.2 Fidelity-Constrained Decoding Framework

During the LLM decoding process, tokens are generated step by step. The next token t_j is then selected based on the input x, prompt, and the previously generated tokens $t_{<j}$. Here, we use the greedy sampling strategy as an example, where the LLM selects the token with the highest probability at each step:

$$t_j = \max_{0 \leq i < |V|} p_\theta(t_i | x, \text{prompt}, t_{<j}), \tag{2}$$

where where θ is the parameters of the LLM and $|V|$ is the vocabulary size.

Our proposed **F**idelity-**C**onstrained **D**ecoding (FCD) affects the decoding process of the LLM by adding the fidelity constraints. The number of valid tokens at each step gradually decreases as the decoding process progresses. In our framework, the LLM determines t_j through

$$t_j = \max_{0 \leq i < |V_j|} p_\theta(t_i | x, \text{prompt}, t_{<j}), \tag{3}$$

where $|V_j|$ is the number of valid tokens in j-th step decoding.

In the FCD framework, the fidelity constraints encompass two aspects: fidelity to legal elements and fidelity to case facts. In the following sections, we will discuss the specific fidelity constraints implemented for several typical legal tasks.

2.3 Legal Judgment Prediction

Fidelity to legal elements. In LJP, LLM is required to output the charge given the legal case facts, and this charge must be a valid charge (i.e., one type of the legal element) from the charge library D_Y, such as theft, traffic accident, etc. Therefore, we treat these charges as the list of candidate tokens. When the LLM outputs the first token, we truncate the candidate token set from the size V (the size of the vocabulary) to $|D_Y|$ (the size of the charge library). For each token generated thereafter, we constrain the LLM to select the token corresponding to the highest probability from the remaining list of charge tokens.

Fidelity to case facts. To enhance fidelity to the case facts, we propose utilizing the case facts to retrieve from the training dataset cases with facts and use the corresponding charges to constrain the generation of the first token. Specifically, we use BM25 (Robertson et al., 1995) to retrieve the top-K candidate charges and rank them accordingly. As we traverse the list of charges from the beginning, if the token is among the top-K charges, we directly select the corresponding charge token list, constraining the model to generate only the remaining tokens for that charge:

$$t_j = \begin{cases} t_{i,0}, & \text{if } t_0 = \{t_{i,0} \mid i < K\}, \\ t_{i,j}, & \text{when } 0 < j < N_i, \end{cases} \tag{4}$$

where N_i is the number of the tokens in i-th charge token list.

2.4 Legal Document Proofreading

Fidelity to legal elements. In LDP, the model is required to output a corrected version of the original document, and most of the text should remain consistent with the original text. We first use the tool spaCy[1] to identify legal-related text in the original document and obtain the positions $P = \{p_0, p_1 ...\}$ of these texts corresponding to tokens $T_l = \{l_0, l_1, ...\}$. When the LLM generates tokens at these positions, we constrain the LLM to only generate the token that corresponds to the original document at that position, effectively setting the probability of all other tokens to zero.

Fidelity to case facts. Legal workers may introduce errors into the documents due to grammar. Therefore, in the process of proofreading legal documents, we refer to the original input document x and impose similarity constraints on the document. Specifically, when generating the i-th token, we decode the top-k tokens from the probability distribution generated by the LLM and check their phonetic and orthographic similarity (Mo, 2024) to the corresponding character at position i in the original text. We select the token with the highest similarity as the proofread token. Therefore, for LTD, the j-th token t_j we generate is:

$$t_j = \begin{cases} l_z, & \text{if } j \in P \wedge j = p_z, \\ \max\limits_{0 \le i < k} \text{sim}(x_j, t_i), & \text{otherwise.} \end{cases} \tag{5}$$

2.5 Legal Trigger Words Detection

Fidelity to legal elements. In LTD, the model needs to identify trigger words for legal events from the original case facts. In the legal case, the trigger words are primarily composed of verbs (i.e., one type of the legal element), as they usually describe the actions or events that lead to legal disputes. Therefore, we use two tools, spaCy and pkuseg (Luo et al., 2019), to recall a set of candidate trigger words from the input factual descriptions of legal cases by identifying verbs. These trigger words are then converted into tokens, forming the candidate token set. The LLM is constrained to only generate these tokens.

Fidelity to case facts. Since there can be repeated trigger words in a case, we also count the occurrences of these trigger words in the original text and incorporate these counts into constraints. If a certain token reaches its maximum allowed occurrences, it will be removed from the subsequent token list. Therefore, the j-th token is generated according to Eq. 3.

Table 1: Statistics of used datasets.

Dataset	#Train	#Test	Avg_Len_I	Avg_Len_O
LJP	1120	560	402.32	8.33
LDP	2983	840	66.49	66.30
LTD	35996	2000	67.57	8.28

3 Experiments

3.1 Experimental Settings

3.1.1 Dataset and Metric

LJP. We conduct experiments using the same dataset as (Shui et al., 2023) from CAIL (Xiao et al., 2018). We use Accuracy (Acc), Precision (P), Recall (R), and F_1-Score (F_1) to evaluate the prediction results of the charges (Feng et al., 2022; Zhong et al., 2018). **LDP.** We use a dataset from Tailing[2]. We use Char-level Precision (P) and $F_{0.5}$ as metrics (Xu et al., 2022). Moreover, considering the LLM over-correction problem (Fang et al., 2023; Li et al., 2023b), we also use False Positives (FP) (Zhang et al., 2022) as an evaluation metric. **LTD.** We use a dataset from LEVEN (Yao et al., 2022) to complete the legal trigger words detection task. We use Precision (P), Recall (R), and F_1-Score (F_1) to evaluate the detection results of trigger words (Xu et al., 2023). The statistics of the above datasets is shown in Table 1. Details are in Appendix B.

3.1.2 Base LLM

We use Qwen (Bai et al., 2023) and Baichuan (Yang et al., 2023) as our base LLMs without any fine-tuning. Specifically, we use Qwen1.5-7B-Chat and Baichuan2-7B-Chat for evaluation and retrieve task examples through BM25 (Robertson et al., 1995) and Sentence-BERT (SBERT) (Reimers, 2019). Additionally, we compared our approach with three legal LLMs: fuzi.mingcha (Wu et al., 2023a), DISC-LawLLM (Yue et al., 2023) and LexiLaw[3].

3.1.3 Implementation Details

Considering resource consumption, we opted to retrieve one example. In the task of legal judgment prediction, we set K in Sec. 2.3 to 20. In the task of legal document proofreading, we set k in Sec. 2.4 to 10. In this paper, we set `do_sample` as False, allowing the LLM to adopt a greedy search strategy and ensuring the reproducibility. All experiments were conducted on Nvidia A6000 GPUs. More details, the source code and datasets can be found at `https://anonymous.4open.science/r/FCD-C3BB`.

[1] https://github.com/explosion/spaCy

[2] https://github.com/DUTIR-LegalIntelligence/Tailing

[3] https://github.com/CSHaitao/LexiLaw

Table 2: Results of LJP, LDP, and LTD. ↓ means smaller is better, in other cases, the larger the value. The best performance is highlighted in bold. **Legal LLM** indicates the LLM has been fine-tuned on extensive legal dataset.

		LJP				**LDP**			**LTD**		
LLM Type	**Model**	**Acc**	**P**	**R**	**F_1**	**FP↓**	**P**	**$F_{0.5}$**	**P**	**R**	**F_1**
Legal	fuzi.mingcha	22.86	44.44	22.65	27.08	1622	32.61	34.72	8.16	14.40	9.04
	LexiLaw	16.61	24.39	16.46	16.89	1315	39.09	40.92	18.14	12.27	13.02
	DiscLaw-LLM	52.86	69.72	52.39	55.20	1119	23.72	23.06	31.81	27.79	28.27
Open-domain	**Qwen1.5-7B-Chat**										
	Zero-shot	29.82	43.74	29.56	30.84	4380	14.77	17.07	47.83	51.69	46.62
	Zero-shot w/ FCD	**51.61**	**56.77**	**51.15**	**49.02**	**1936**	**25.11**	**27.00**	**56.65**	**59.52**	**54.29**
	Few-shot-BM25	41.79	63.85	41.42	45.53	2823	24.64	27.70	57.85	63.09	56.77
	Few-shot-BM25 w/ FCD	**58.93**	**64.72**	**58.93**	**57.17**	**1113**	**42.95**	**44.19**	**61.34**	**64.56**	**59.30**
	Few-shot-SBERT	37.14	55.72	36.81	40.14	2812	25.80	29.04	55.60	61.19	54.84
	Few-shot-SBERT w/ FCD	**55.71**	**61.19**	**55.71**	**54.18**	**1175**	**39.59**	**40.71**	**59.17**	**62.99**	**57.43**
	Baichuan2-7B-Chat										
	Zero-shot	28.39	39.96	28.14	29.31	1710	25.03	26.43	7.94	6.10	6.62
	Zero-shot w/ FCD	**43.75**	**50.80**	**43.34**	**41.32**	**624**	**31.05**	**26.52**	**31.41**	**21.29**	**23.52**
	Few-shot-BM25	36.61	61.44	36.28	41.61	1395	41.11	43.66	52.68	52.86	49.45
	Few-shot-BM25 w/ FCD	**56.61**	**64.82**	**56.61**	**55.11**	**722**	**52.81**	**51.81**	**56.83**	52.71	**51.18**
	Few-shot-SBERT	33.21	53.65	32.92	36.85	1382	40.42	42.32	48.80	46.94	44.68
	Few-shot-SBERT w/ FCD	**53.93**	**57.81**	**53.45**	**51.17**	**677**	**48.67**	**46.16**	**55.95**	**49.68**	**48.89**

Table 3: The efficiency analysisof FCD.

Task	**Total (s)**	**Per-token (ms)**	**F-value**
LJP	145.10	54.26	30.84
LJP w/ FCD	143.13	55.19	49.02
LDP	1553.52	41.65	17.07
LDP w/ FCD	1609.05	44.46	27.00
LTD	700.89	48.91	46.62
LTD w/ FCD	561.25	49.28	54.29

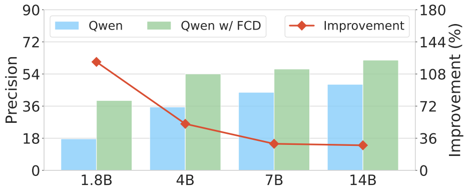

Figure 3: Performances of FCD with various LLM sizes.

3.2 Results and Analysis

3.2.1 Overall Performance.

The main experimental results are shown in Table 2. Key findings include: 1) FCD outperforms few-shot: Unconstrained LLMs, even with examples, perform worse than adding FCD directly. 2) FCD is compatible with RAG settings: Adding FCD on top of few-shot methods consistently leads to improvements. This indicates that FCD can be integrated with retrieval systems to further enhance the model's performance. 3) FCD surpasses legal LLMs: Without fine-tuning, an open-domain LLM with one retrieved example and FCD achieves performance comparable to legal LLMs, highlighting FCD's superiority.

3.2.2 Efficiency Analysis

We used Qwen in the zero-shot setting to explore the efficiency of FCD in three tasks on a Nvidia A6000 GPU, with the results shown in Table 3. In terms of total decoding time (**Total**), FCD achieved shorter total decoding times in both the LJP and LTD tasks. This is because FCD reduces the decoding space, enabling the LLM to focus more quickly on outputs that align with legal elements and case facts, thereby reducing the overall decoding time. For average decoding time per token (**Per-token**), the addition of FCD results in a time efficiency within 3 ms. However, given the significant performance improvement, the slight increase in time is almost negligible.

3.2.3 Improvement Across Various LLM Sizes

We investigate how FCD performs across different LLM sizes (Qwen: 1.8B, 4B, 7B, 14B) on the LJP task, in a zero-shot manner. The results are shown in Fig. 3. We can see that adding FCD to LLMs of different sizes consistently improves performance, reflecting the universality and effectiveness of FCD. Furthermore, we find that adding FCD to smaller-scale LLMs can even surpass larger-scale LLMs without constraints (e.g., "4B w/ FCD" performs better than "14B"). This reflects that FCD can introduce additional legal-specific knowledge and reasoning abilities into smaller models, which even larger models might not have learned without FCD.

4 Conclusion

This paper introduces a novel research perspective to tackle the fidelity and efficiency issues in legal tasks by constraining the decoding process of LLMs. Our FCD explicitly ensures that only outputs aligned with case facts and legal elements are generated. Experiments on three datasets using two open-domain LLMs in zero-shot and RAG settings demonstrate FCD's effectiveness and adaptability.

5 Limitations

Although FCD has achieved superior performance, it still has the following limitations:

First, FCD is a general framework that helps LLMs adapt to various legal tasks without any fine-tuning, but we have not explored its application across a broader range of legal tasks. Therefore, future work could involve extending FCD to more legal tasks, such as legal named-entity recognition (Leitner et al., 2019) and article prediction (Xiao et al., 2018). Additionally, when applying LLMs to legal tasks, hallucinations (Dahl et al., 2024; Li, 2023) may arise, and exploring how FCD can mitigate hallucinations is another potential direction for future research.

Furthermore, this paper explores the combination of FCD with the retrieval system to jointly enhance the LLM's ability to perform legal tasks. However, we only selected two commonly used retrieval methods, BM25 and Sentence-BERT. Future research could explore the integration of more advanced retrieval-augmented generation (RAG) methods and retrieval models specifically designed for the legal domain, such as Lawformer (Xiao et al., 2021).

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Andrew Blair-Stanek, Nils Holzenberger, and Benjamin Van Durme. 2023. Can gpt-3 perform statutory reasoning? In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, pages 22–31.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

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 preprint arXiv:2306.16092*.

Matthew Dahl, Varun Magesh, Mirac Suzgun, and Daniel E Ho. 2024. Large legal fictions: Profiling legal hallucinations in large language models. *Journal of Legal Analysis*, 16(1):64–93.

Wentao Deng, Jiahuan Pei, Keyi Kong, Zhe Chen, Furu Wei, Yujun Li, Zhaochun Ren, Zhumin Chen, and Pengjie Ren. 2023a. Syllogistic reasoning for legal judgment analysis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13997–14009.

Wentao Deng, Jiahuan Pei, Keyi Kong, Zhe Chen, Furu Wei, Yujun Li, Zhaochun Ren, Zhumin Chen, and Pengjie Ren. 2023b. Syllogistic reasoning for legal judgment analysis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13997–14009, Singapore. Association for Computational Linguistics.

Qian Dong and Shuzi Niu. 2021. Legal judgment prediction via relational learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 983–992.

Tao Fang, Shu Yang, Kaixin Lan, Derek F Wong, Jinpeng Hu, Lidia S Chao, and Yue Zhang. 2023. Is chatgpt a highly fluent grammatical error correction system? a comprehensive evaluation. *arXiv preprint arXiv:2304.01746*.

Zhiwei Fei, Xiaoyu Shen, Dawei Zhu, Fengzhe Zhou, Zhuo Han, Songyang Zhang, Kai Chen, Zongwen Shen, and Jidong Ge. 2023. Lawbench: Benchmarking legal knowledge of large language models. *arXiv preprint arXiv:2309.16289*.

Yi Feng, Chuanyi Li, and Vincent Ng. 2022. Legal judgment prediction: A survey of the state of the art. In *IJCAI*, pages 5461–5469.

Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. 2023. Grammar-constrained decoding for structured nlp tasks without finetuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10932–10952.

James Y Huang, Sailik Sengupta, Daniele Bonadiman, Yi-an Lai, Arshit Gupta, Nikolaos Pappas, Saab Mansour, Katrin Kirchoff, and Dan Roth. 2024. Deal: Decoding-time alignment for large language models. *arXiv preprint arXiv:2402.06147*.

Cong Jiang and Xiaolei Yang. 2023. Legal syllogism prompting: Teaching large language models for legal judgment prediction. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, pages 417–421.

Sergey Kuleshov, Alexandra Zaytseva, and Konstantin Nenausnikov. 2020. Legal tech: Documents' validation method based on the associative-ontological approach. In *International Conference on Speech and Computer*, pages 244–254. Springer.

Elena Leitner, Georg Rehm, and Julian Moreno-Schneider. 2019. Fine-grained named entity recognition in legal documents. In *International conference on semantic systems*, pages 272–287. Springer.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori B Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023a. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12286–12312.

Yinghui Li, Haojing Huang, Shirong Ma, Yong Jiang, Yangning Li, Feng Zhou, Hai-Tao Zheng, and Qingyu Zhou. 2023b. On the (in) effectiveness of large language models for chinese text correction. *arXiv preprint arXiv:2307.09007*.

Zihao Li. 2023. The dark side of chatgpt: Legal and ethical challenges from stochastic parrots and hallucination. *arXiv preprint arXiv:2304.14347*.

Jinlong Liu and Xudong Luo. 2024. A bert-based model for legal document proofreading. In *International Conference on Intelligent Information Processing*, pages 190–206. Springer.

Jinliang Lu, Chen Wang, and Jiajun Zhang. 2024. Diver: Large language model decoding with span-level mutual information verification. *arXiv preprint arXiv:2406.02120*.

Ruixuan Luo, Jingjing Xu, Yi Zhang, Zhiyuan Zhang, Xuancheng Ren, and Xu Sun. 2019. Pkuseg: A toolkit for multi-domain chinese word segmentation. *CoRR*, abs/1906.11455.

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*, pages 2342–2348.

Yongzhuo Mo. 2024. char-similar. https://github.com/yongzhuo/char-similar.

John J Nay. 2023. Large language models as fiduciaries: a case study toward robustly communicating with artificial intelligence through legal standards. *arXiv preprint arXiv:2301.10095*.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at trec-3. *Nist Special Publication Sp*, 109:109.

Jaromir Savelka, Kevin D Ashley, Morgan A Gray, Hannes Westermann, and Huihui Xu. 2023. Explaining legal concepts with augmented large language models (gpt-4). *arXiv preprint arXiv:2306.09525*.

Ruihao Shui, Yixin Cao, Xiang Wang, and Tat-Seng Chua. 2023. A comprehensive evaluation of large language models on legal judgment prediction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7337–7348.

Shiguang Wu, Zhongkun Liu, Zhen Zhang, Zheng Chen, Wentao Deng, Wenhao Zhang, Jiyuan Yang, Zhitao Yao, Yougang Lyu, Xin Xin, Shen Gao, Pengjie Ren, Zhaochun Ren, and Zhumin Chen. 2023a. fuzi.mingcha. `https://github.com/irlab-sdu/fuzi.mingcha`.

Yiquan Wu, Siying Zhou, Yifei Liu, Weiming Lu, Xiaozhong Liu, Yating Zhang, Changlong Sun, Fei Wu, and Kun Kuang. 2023b. Precedent-enhanced legal judgment prediction with llm and domain-model collaboration. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12060–12075.

Chaojun Xiao, Xueyu Hu, Zhiyuan Liu, Cunchao Tu, and Maosong Sun. 2021. Lawformer: A pre-trained language model for chinese legal long documents. *AI Open*, 2:79–84.

Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Yansong Feng, Xianpei Han, Zhen Hu, Heng Wang, et al. 2018. Cail2018: A large-scale legal dataset for judgment prediction. *arXiv preprint arXiv:1807.02478*.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, and Enhong Chen. 2023. Large language models for generative information extraction: A survey. *arXiv preprint arXiv:2312.17617*.

Lvxiaowei Xu, Jianwang Wu, Jiawei Peng, Jiayu Fu, and Ming Cai. 2022. Fcgec: Fine-grained corpus for chinese grammatical error correction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1900–1918.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

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*, pages 183–201.

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

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*.

Yue Zhang, Zhenghua Li, Zuyi Bao, Jiacheng Li, Bo Zhang, Chen Li, Fei Huang, and Min Zhang. 2022. Mucgec: a multi-reference multi-source evaluation dataset for chinese grammatical error correction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3118–3130.

Zheng Zhao, Emilio Monti, Jens Lehmann, and Haytham Assem. 2024. Enhancing contextual understanding in large language models through contrastive decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4225–4237.

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*, pages 3540–3549.

A Related Work

A.1 LLM for Legal Tasks

Recently, numerous studies have applied LLMs to the legal domain (Yu et al., 2022; Savelka et al., 2023; Wu et al., 2023b). Prompt-based methods (Blair-Stanek et al., 2023; Nay, 2023; Jiang and Yang, 2023) enrich the prompts by retrieving similar examples to activate the LLM's legal knowledge (Shui et al., 2023; Deng et al., 2023a). Additionally, Legal LLMs (Yue et al., 2023; Cui et al., 2023; Wu et al., 2023a; Deng et al., 2023b) have also emerged, which, after fine-tuning on extensive legal data, have acquired the capability to engage in basic legal dialogues. Unlike previous works, this paper aims to provide a method that requires no fine-tuning at all, adapting open-domain LLMs to legal scenarios, which can be conveniently transferred to any new legal task.

A.2 Constrained Decoding

In recent years, several works have proposed the use of constrained decoding to guide the generation process of LLMs, ensuring that the output meets expected standards. (Geng et al., 2023) introduced grammar-constrained decoding to enable LLMs to perform structured NLP tasks. (Lu et al., 2024) improved the quality of LLM outputs by introducing span-level pointwise mutual information scores during the decoding process to select optimal spans. Contrastive decoding constraints (Zhao et al., 2024; Li et al., 2023a) have also been used to enhance LLM's text generation capabilities. (Huang et al., 2024) aligns LLMs during decoding to produce content that aligns with human preferences. Unlike previous works, our approach starts from the need for high fidelity in legal scenarios. We have designed decoding constraints that are faithful to the legal elements and the case facts, ensuring that the LLM's outputs adhere to these fidelity constraints.

B More Details of Evaluated Models and Datasets

Table 4 is the website URLs and corresponding licenses of the evaluated models and datasets. The datasets we use have all been anonymized.

Type	Dataset/LLM	URL	Licence
Dataset	CAIL2018 (LJP)	https://github.com/china-ai-law-challenge/CAIL2018	MIT License
	Tailing (LDP)	https://github.com/DUTIR-LegalIntelligence/Tailing	
	LEVEN (LTD)	https://github.com/thunlp/LEVEN	
LLM	fuzi.mingcha	https://github.com/irlab-sdu/fuzi.mingcha	Apache-2.0 license
	DISC-LawLLM	https://github.com/FudanDISC/DISC-LawLLM	Apache-2.0 license
	LexiLaw	https://github.com/CSHaitao/LexiLaw	MIT license
	Qwen1.5-7B-Chat	https://huggingface.co/Qwen/Qwen1.5-7B-Chat	Apache-2.0 license
	Baichuan2-7B-Chat	https://github.com/baichuan-inc/Baichuan2	Apache-2.0 license

Table 4: The dataset source URLs and licenses. The parts where the license is listed as empty indicate that the author has not provided a License.