LAWGPT: KNOWLEDGE-GUIDED DATA GENERATION AND ITS APPLICATION TO LEGAL LLM

Zhi Zhou
Nanjing University

Kun-Yang Yu
Nanjing University

Shi-Yu Tian
Nanjing University

Xiao-Wen Yang
Nanjing University

Jiang-Xin Shi
Nanjing University

Peng-Xiao Song
Nanjing University

Yi-Xuan Jin
Nanjing University

Lan-Zhe Guo & Yu-Feng Li
Nanjing University
{guolz,liyf}@nju.edu.cn

ABSTRACT

Large language models (LLMs), both proprietary and open-source, have demonstrated remarkable capabilities across various natural language processing tasks. However, they face significant limitations in legal reasoning tasks. Proprietary models introduce data privacy risks and high inference costs, while open-source models underperform due to insufficient legal domain training data. To this end, we study data generation for legal reasoning to improve the performance of open-source legal LLMs with the help of proprietary LLMs, which is challenging due to the lack of legal knowledge in open-source LLMs and the difficulty in verifying the generated data. To address these challenges, we propose KGDG, a knowledge-guided data generation framework for legal reasoning. Our framework enables leveraging legal knowledge to enhance generation diversity and introduces a refinement and verification process to ensure the quality of generated data. Moreover, we expand the generated dataset to further enhance the LLM reasoning capabilities. Using KGDG, we create a synthetic legal reasoning dataset containing 50K high-quality examples. Our trained model LAWGPT outperforms existing legal-specific LLMs and achieves performance comparable to proprietary LLMs, demonstrating the effectiveness of both KGDG and LAWGPT. Both our code and resources is publicly available at https://anonymous.4open.science/r/KgDG-45F5.

1 INTRODUCTION

Large language models (LLMs) (OpenAI, 2023b; Touvron et al., 2023) have achieved remarkable success in various natural language processing (NLP) tasks, including natural language understanding (Dong et al., 2019), reasoning (Huang & Chang, 2023), and generation (Yu et al., 2022). Both proprietary and open-source LLMs exhibit strong generalization capabilities, enabling their application in diverse downstream scenarios, such as medicine (Thirunavukarasu et al., 2023), finance (Yang et al., 2023), education (Gan et al., 2023). Recent studies (Fei et al., 2023; Nguyen, 2023) have demonstrated the preliminary effectiveness of existing general LLMs in legal reasoning tasks, including legal documents retrieval (Chen et al., 2013), legal judgment prediction (Luo et al., 2017), and legal question answering (Zhong et al., 2020).

Despite their preliminary success in legal applications, LLMs still have significant limitations in practical legal reasoning tasks. Proprietary LLMs such as GPT-4 (OpenAI, 2023b) and GPT-3.5 Turbo (OpenAI, 2023a) require API access, which introduces significant data privacy risks and high inference costs. Open-source LLMs like Qwen (Yang et al., 2024) and ChatGLM (Du et al., 2022) demonstrate suboptimal performance due to insufficient legal domain training data. These limitations present an opportunity to leverage proprietary LLMs for generating synthetic legal reasoning data to build open-source legal LLMs.

Existing studies have developed various LLM-based data generation methods for downstream reasoning tasks, such as mathematical reasoning (Luo et al., 2025). These methods assume that the LLMs used for generation possess sufficient knowledge about the downstream tasks and can generate diverse data through appropriate prompts (Yu et al., 2024). Moreover, for math reasoning, the formal nature makes it straightforward to verify synthetic data (Li et al., 2024b) and eliminate incorrect data caused by LLM hallucination. However, legal reasoning presents unique challenges: general LLMs lack specific legal knowledge, which limits the diversity of generation. Additionally, the complex nature of legal reasoning makes it difficult to formalize and verify the generated data.

To address these challenges, we propose KGDG, a knowledge-guided data generation framework for legal reasoning tasks. Our framework consists of three key components: (1) Knowledge-Guide Generation, which leverages a legal knowledge base $\mathcal{K}$ to generate diverse data; (2) Knowledge-Guide Fixer, which refines incorrect references and reasoning paths; and (3) Data Verifier, which filters out uncorrectable data to ensure quality. To further enhance the reasoning capabilities of trained LLMs, we propose a Mixture Training strategy that expands the generated dataset. Using KGDG, we create a synthetic legal reasoning dataset containing 50K high-quality examples. Our trained model LawGPT outperforms existing legal-specific LLMs and achieves performance comparable to proprietary LLMs, demonstrating the effectiveness of both KGDG and LawGPT. Our contributions can be summarized as follows:

(a) We propose KGDG, a knowledge-guided data generation framework that enables the creation of high-quality and diverse datasets for legal reasoning tasks.
(b) We create a large-scale synthetic dataset using KGDG and train LawGPT with different model scales. The dataset and models will be publicly available to facilitate future research.
(c) We demonstrate through extensive experiments that LawGPT outperforms state-of-the-art legal-specific LLMs and achieves performance comparable to proprietary LLMs in legal reasoning tasks.

2 Methodology

In this section, we introduce KGDG, an LLM-based data generation framework, building data to improve the legal reasoning performance of domain-specific LLMs. However, the following two challenges make it difficult for general LLMs to generate data for legal reasoning:

(a) General LLMs lack domain-specific legal knowledge, which limits the diversity and quality of synthetic data generation.
(b) Legal synthetic data is difficult to formalize and verify, making it challenging to detect and eliminate hallucinations in the generation process.

We design *Knowledge-Guided Generation* (KgGen) to address the first challenge by introducing domain specific legal documents. Then, *Knowledge-Guided Fixer* (KgFix) and *Data Verifier* (DaVer) addressing the second challenge by refining correctable errors and removing uncorrectable data. To further improve model reasoning performance, we implement a *Mixture Training* (MiTra) to teach domain-specific LLMs to reason step-by-step while keeping the capability to generate direct answers efficiently. Overall illustration is shown in Figure 1 and each module is detailed below.

2.1 Knowledge-Guided Generation (KgGen)

Existing studies (Li et al., 2024b) demonstrate that LLM-based data generation methods have strong potential for building high-quality training data. However, for tasks that require specific domain knowledge, such as legal reasoning, general LLMs may fail to build high-quality data due to their lack of domain knowledge, leading to insufficient diversity in synthetic data. To address this challenge, we design KgGen by introducing a knowledge base $\mathcal{K}$ to compensate for the lack of legal knowledge in general LLMs. This enables us to expand the diversity of synthetic data through professional legal knowledge sampling.

Specifically, for legal reasoning task, KgGen consists of two components: *Knowledge-Aware Sampler* and *Knowledge-Guided Writer*. The *Knowledge-Aware Sampler* employs sampling strategies to enhance the diversity of synthetic data, while the *Knowledge-Guided Writer* leverages general LLMs to extract core information and generate question-answer pairs.

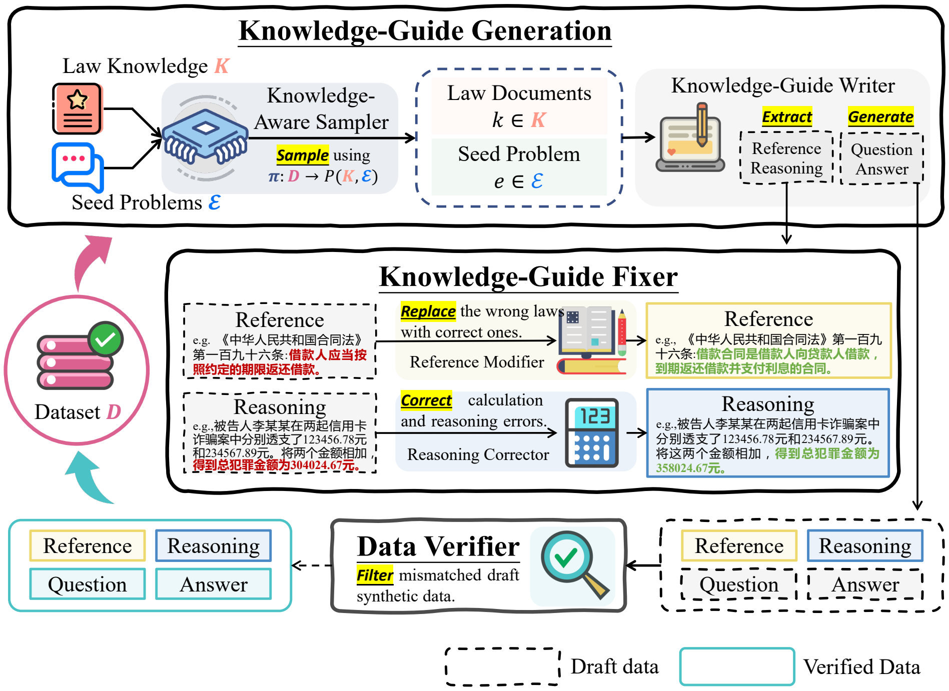
Knowledge-Guide Generation
Law Knowledge K
Seed Problems $\mathcal{E}$
Knowledge-Aware Sampler
Sample using
$\pi: D \rightarrow P(K, \mathcal{E})$
Law Documents
$k \in K$
Seed Problem
$e \in \mathcal{E}$
Knowledge-Guide Writer
Extract
Generate
Reference
Reasoning
Question
Answer
Dataset D
Knowledge-Guide Fixer
Reference
e.g. 《中华人民共和国合同法》第一百九十六条:借款人应当按照约定的期限返还借款。
Replace the wrong laws with correct ones.
Reference Modifier
Reference
e.g., 《中华人民共和国合同法》第一百九十六条:借款合同是借款人向贷款人借款，到期返还借款并支付利息的合同。
Reasoning
e.g.,被告人李某某在两起信用卡诈骗案中分别透支了123456.78元和234567.89元。将两个金额相加，得到总犯罪金额为304024.67元。
Correct calculation and reasoning errors.
Reasoning Corrector
Reasoning
e.g.,被告人李某某在两起信用卡诈骗案中分别透支了123456.78元和234567.89元。将这两个金额相加，得到总犯罪金额为358024.67元。
Reference
Reasoning
Question
Answer
Data Verifier
Filter mismatched draft synthetic data.
Reference
Reasoning
Question
Answer
Draft data
Verified Data

Figure 1: Illustration of KGDG, a LLM-based data generation framework.

Knowledge-Aware Sampler takes two inputs: a knowledge base $\mathcal{K}$ containing legal documents and a seed problem set $\mathcal{E}$ providing format examples for legal reasoning tasks. The sampling process is controlled by a strategy $\pi(\mathbf{k}, \mathbf{e}|\mathcal{D}_{\text{Gen}})$ that samples from $\mathcal{K}$ and $\mathcal{E}$ conditioned on the current generated dataset $\mathcal{D}_{\text{Gen}}$, where $\mathbf{k} \in \mathcal{K}$ represents a sampled legal document and $\mathbf{e} \in \mathcal{E}$ represents a sampled seed problem. We implement π as a two-step sampling strategy: (1) LLM selects specific types of legal knowledge from $\mathcal{K}$ based on the sampled example problem $\mathbf{e}$ to ensure consistency between the example and knowledge; (2) Monte Carlo sampling ensures diverse and balanced synthetic data across all problem types and their corresponding legal knowledge domains.

The *Knowledge-Guided Writer* LLM_{W} takes the sampled legal document $\mathbf{k}$ and example problem $\mathbf{e}$ as input, generating the unverified draft data $\tilde{\mathbf{x}}$ containing question $\tilde{\mathbf{q}}$, answer $\tilde{\mathbf{a}}$, reasoning path $\tilde{\mathbf{p}}$, and references $\tilde{\mathbf{r}}$:

$$\tilde{\mathbf{x}} = (\tilde{\mathbf{q}}, \tilde{\mathbf{a}}, \tilde{\mathbf{r}}, \tilde{\mathbf{p}}) = \text{LLM}_{\text{W}}(\mathbf{k}, \mathbf{e}) \tag{1}$$

2.2 KNOWLEDGE-GUIDE FIXER (KGFIX) AND DATA VERIFIER (DAVER)

The unverified draft data $\tilde{\mathbf{x}} = (\tilde{\mathbf{q}}, \tilde{\mathbf{a}}, \tilde{\mathbf{r}}, \tilde{\mathbf{p}})$ contains potential errors in all components due to the hallucination problems of general LLMs. To address this issue, we introduce KGFIX to fix correctable errors in the reasoning path $\tilde{\mathbf{p}}$ and references $\tilde{\mathbf{r}}$, and DAVER to filter out uncorrectable data.

KGFIX consists of two components: *Reference Modifier* and *Reasoning Corrector*. The *Reference Modifier* replaces LLM-generated legal references with verified references from the knowledge base, producing a corrected reference $\hat{r} = \text{Fixer}_{\text{M}}(\tilde{r}, \mathcal{K})$. The *Reasoning Corrector* refines the reasoning path by removing logical errors and ensuring consistency, generating a corrected reasoning path $\hat{p} = \text{Fixer}_{\text{C}}(\tilde{p})$.

While KGFIX ensures the correctness of reference $\hat{r}$ and reasoning path $\hat{p}$, it cannot guarantee their relevance to the generated question-answer pair. Therefore, we implement DAVER to validate whether the answer $\tilde{a}$ can be derived from the question $\tilde{q}$ using the corrected references $\hat{r}$ and reasoning path $\hat{p}$. If the validation succeeds, we mark the question-answer pair as valid (denoted as $\hat{q}$ and $\hat{a}$). The verified data $\hat{x} = (\hat{q}, \hat{a}, \hat{r}, \hat{p})$ is then added to the synthetic dataset $\mathcal{D}_{\text{Gen}}$. This process continues until $|\mathcal{D}_{\text{Gen}}|$ meets the required data volume.

2.3 MIXTURE TRAINING (MITRA)

To further enhance the reasoning performance of the trained LLM, we implement MITRA to generate two types of training data using $\mathcal{D}_{\text{Gen}}$: (1) standard question-answer pairs and (2) question-answer pairs with explicit reasoning paths. The standard pairs enable efficient direct responses, while the pairs with reasoning paths teach the model step-by-step reasoning.

Specifically, we design two prompt templates: $\mathrm{T}_s(\hat{\mathbf{q}}, \hat{\mathbf{a}})$ for standard pairs and $\mathrm{T}_r(\hat{\mathbf{q}}, \hat{\mathbf{a}}, \hat{\mathbf{r}}, \hat{\mathbf{p}})$ for pairs with reasoning paths. Here, T_s generates training instances using only questions and answers, while T_r incorporates additional reasoning paths $\hat{\mathbf{r}}$ and legal references $\hat{\mathbf{p}}$. The final training dataset is constructed by combining both types:

$$\mathcal{D}_{\text{Train}} = \{\mathrm{T}_s(\hat{\mathbf{q}}, \hat{\mathbf{a}})\} \cup \{\mathrm{T}_r(\hat{\mathbf{q}}, \hat{\mathbf{a}}, \hat{\mathbf{r}}, \hat{\mathbf{p}})\}, \quad (\hat{\mathbf{q}}, \hat{\mathbf{a}}, \hat{\mathbf{r}}, \hat{\mathbf{p}}) \sim \mathcal{D}_{\text{Gen}} \tag{2}$$

3 EXPERIMENTS

In this section, we compare the performance of LAWGPT against baseline models and law-specific models to demonstrate the effectiveness of both our KGDG framework and the trained LAWGPT.

3.1 EXPERIMENTAL SETTINGS

Evaluation Protocol. To evaluate the legal reasoning performance of each model, we adopt four legal reasoning tasks: Scene-based Article Prediction (Task #1) (Liu et al., 2023), Prison Term Prediction without Article (Task #2), Prison Term Prediction with Article (Task #3) (Xiao et al., 2018), and Criminal Damages Calculation (Task #4) [1]. Task #1 is evaluated using the ROUGE-L score to compare the legal article prediction with the ground truth. Tasks #2 and #3 are evaluated using Normalized log-distance to compare the predicted prison term. Task #4 is evaluated using accuracy to determine whether the predicted damages match the ground truth.

Comparison Models. We compare two types of models: (1) General proprietary LLMs, including GPT-4 (OpenAI, 2023b), GPT-3.5 Turbo (OpenAI, 2023a), and DeepSeek V3 (DeepSeek-AI et al., 2024); (2) Law-specific LLMs, including Lexilaw (Li et al., 2024a), LaywerLLaMA (Huang et al., 2023), HanFei (He et al., 2023), ChatLaw (Cui et al., 2023), FuziMingcha (Deng et al., 2023), and WisdomInterrogatory (Wu et al., 2024).

Dataset Construction. We implement the KGDG framework using the DeepSeek V3 model (DeepSeek-AI et al., 2024), based on a legal knowledge base and seed problems. Specifically, to construct the legal knowledge base, we manually collect 186,197 high-quality criminal legal documents and 152,452 civil legal documents. Each document includes judgment facts, reasons, results, and relevant laws. This knowledge base supports the generation of diverse and synthetic problems for legal reasoning, as well as the verification and correction of generated reasoning paths and answers. For seed problems, we manually construct ten problems for each task as examples to guide the KGDG to generate legal problems in the desired format. These seed problems are solely for demonstration and are not used for training. KGDG generates 25K legal problems with verified answers. Each problem has two versions: one with generated answers and one with answers and detailed reasoning steps, resulting in about 50K training data in total. The detailed implementation of KGDG and generation process is provided in Appendix A.

Model Training. We adopts the LLaMA-Factory (Zheng et al., 2024) to fine-tune the series of Qwen-2.5 models (Yang et al., 2024), including 0.5B, 1.5B, and 3B version. The training epochs are set to 3 and learning rate is set to 1e-5 with a cosine learning rate scheduler. The implementation of our evluation is based on the LawBench (Fei et al., 2023). Our experiments are conducted on a Linux server with 4 NVIDIA A800 GPUs.

[1] `https://laic.cjbdi.com/`

Table 1: Performance comparison between LAWGPT and Qwen-2.5 models of different model scales. LAWGPT consistently outperforms Qwen-2.5 across all model sizes and tasks, showing the effectiveness of our KGDG framework.

Models	#Parameters	Task #1	Task #2	Task #3	Task #4	Average
Qwen-2.5	0.5B	27.9	81.2	80.1	45.0	58.6
LAWGPT	0.5B	33.1	86.8	86.6	62.0	67.1
Δ Performance		↑ 5.2	↑ 5.6	↑ 6.5	↑ 14.0	↑ 9.5
Qwen-2.5	1.5B	29.9	82.4	82.3	49.0	60.9
LAWGPT	1.5B	35.7	87.4	87.3	68.0	69.6
Δ Performance		↑ 5.8	↑ 5.0	↑ 5.0	↑ 19.0	↑ 8.7
Qwen-2.5	3.0B	28.7	81.7	79.9	56.0	61.6
LAWGPT	3.0B	37.7	88.2	88.0	73.2	71.8
Δ Performance		↑ 9.0	↑ 6.5	↑ 8.1	↑ 17.2	↑ 10.2

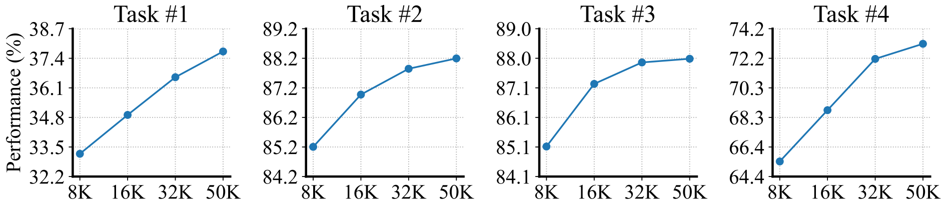

Figure 2: Scalability analysis of the KGDG framework. The performance on all tasks improves as the amount of generated training data increases.

3.2 EMPIRICAL RESULTS

In this section, we conduct experiments to compare the performance of LAWGPT with baseline models and law-specific models to demonstrate the effectiveness of our KGDG framework as well as the trained legal LLM LAWGPT.

Effectiveness of KGDG. To evaluate the effectiveness of our proposed KGDG data generation framework, we fine-tune Qwen-2.5 models of different scales using our generated 50K data. The results in Table 1 demonstrate that out fine-tuned model consistently outperforms the base models across all scales. This indicates that KGDG generates high-quality legal data that effectively improves the reasoning capabilities of base models regardless of their size. Moreover, we analyze the scalability of KGDG in Figure 2. The experimental results demonstrate that the performance of trained LLMs consistently improves across all tasks as the volume of generated training data increases, indicating the strong potential of KGDG for developing more capable legal LLMs.

Effectiveness of LAWGPT. We evaluate LAWGPT against both general and law-specific LLMs. For general LLMs, we include two proprietary models (GPT-4 and GPT-3.5 Turbo) and one large-scale open-source model (DeepSeek V3). We also compare against seven law-specific LLMs of various sizes. As shown in Table 2, LAWGPT outperforms all existing law-specific LLMs despite its smaller scale. Furthermore, LAWGPT surpasses GPT-4 and GPT-3.5 Turbo while achieving performance comparable to DeepSeek V3 on multiple tasks. These results demonstrate both the value of specialized legal LLMs and the effectiveness of our KGDG framework.

3.3 ABLATION STUDY

We conduct an ablation study using a 4K subset of the training data to evaluate the effectiveness of each component in our KGDG framework. The results are shown in Table 3. The model achieves its best average performance only when all four modules are integrated. For Task #2 and #3, we observe that the DAVER module introduces a slight performance degradation when handling complex prison

Table 2: Performance comparison between LawGPT and general LLMs and law-specific LLMs. The results show that LawGPT outperforms exisings law-specific LLMs. Moreover, LawGPT can achieves similar performance to general LLMs even with a significantly smaller scale. The best performance is highlighted in bold and the second best is underlined among law-specific LLMs.

Models	#Parameters	Task #1	Task #2	Task #3	Task #4	Average
General LLMs						
Deepseek V3	671B	38.1	87.5	86.8	84.4	74.2
GPT-4	-	27.5	82.6	81.9	77.6	67.4
GPT-3.5 Turbo	-	31.3	78.7	76.8	61.2	62.0
Law-Specific LLMs						
Lexilaw	7B	35.8	78.1	74.9	35.8	56.1
HanFei	7B	33.6	73.1	69.6	39.4	53.9
FuziMingcha	7B	22.2	77.2	75.5	47.2	55.5
WisdomInterrogatory	7B	32.0	80.4	81.1	17.4	52.7
LaywerLLaMA	13B	25.9	74.2	75.5	39.2	53.7
ChatLaw	13B	31.6	76.2	73.6	41.4	55.7
ChatLaw	33B	26.0	67.0	53.6	41.6	47.1
LawGPT	0.5B	33.1	86.8	86.6	62.0	67.1
LawGPT	1.5B	35.7	87.4	87.3	68.0	69.6
LawGPT	3B	**37.7**	**88.2**	**88.0**	**73.2**	**71.8**

Table 3: Ablation study. We conduct experiments on the Qwen-2.5-3B model using a 4K subset of generated data. Our four proposed modules are added sequentially to assess their effectiveness. The results show that the best average performance is achieved when all four modules are integrated.

KgGen	KgFix	DaVer	MiTra	Task #1	Task #2	Task #3	Task #4	Average
				28.7	81.7	79.9	56.0	61.6
✓				30.9	84.9	84.7	59.8	65.1
✓	✓			31.1	**85.6**	**85.4**	60.4	65.6
✓	✓	✓		31.5	85.1	84.8	65.0	66.6
✓	✓	✓	✓	**33.2**	85.2	85.1	**65.4**	**67.2**

term prediction tasks, indicating potential room for improvement in this module. Nevertheless, the integration of all four modules still yields the best overall performance, demonstrating the value of each component in our KgDG framework.

4 Conclusion

In this paper, we study data generation for legal reasoning to improve the performance of open-source legal LLMs with the help of proprietary LLMs. To address the challenges of insufficient legal knowledge in open-source LLMs and difficulty in verifying generated data, we propose KgDG, a knowledge-guided data generation framework. Our framework consists of three key components that leverage legal knowledge to enhance generation diversity and ensure data quality through refinement and verification processes. Additionally, we develop MiTra to expand the generated dataset and further enhance LLM reasoning capabilities. Both KgDG and LawGPT are validated by extensive experiments on multiple legal reasoning tasks.

Limitations and Future Work. This paper gives a preliminary study on the data generation for legal LLMs and we only make a simple attempt to build each component in the KgDG framework, which is mainly relies on prompting LLMs, and could be further improved by incorporating more sophisticated techniques. Moreover, we only synthet 50K data for training, which is enough for validating the effectiveness of KGDG and LAWGPT, but the upperbound of KGDG could be further explored by generating more data.

REFERENCES

Yen-Liang Chen, Yi-Hung Liu, and Wu-Liang Ho. A text mining approach to assist the general public in the retrieval of legal documents. *Journal of the American Society for Information Science and Technology*, 64(2):280–290, 2013.

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

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, and Wangding Zeng. Deepseek-v3 technical report. *CoRR*, abs/2412.19437, 2024.

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

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, pp. 13042–13054, 2019.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 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)*, pp. 320–335, 2022.

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

Wensheng Gan, Zhenlian Qi, Jiayang Wu, and Jerry Chun-Wei Lin. Large language models in education: Vision and opportunities. In *Proceedings of the IEEE International Conference on Big Data*, pp. 4776–4785, 2023.

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

Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey. In *Findings of the Association for Computational Linguistics*, pp. 1049–1065, 2023.

Quzhe Huang, Mingxu Tao, Zhenwei An, Chen Zhang, Cong Jiang, Zhibin Chen, Zirui Wu, and Yansong Feng. Lawyer llama technical report. *CoRR*, abs/2305.15062, 2023.

Haitao Li, Qingyao Ai, Qian Dong, and Yiqun Liu. Lexilaw: A scalable legal language model for comprehensive legal understanding, 2024a. URL `https://github.com/CSHaitao/LexiLaw`.

Zenan Li, Zhi Zhou, Yuan Yao, Xian Zhang, Yu-Feng Li, Chun Cao, Fan Yang, and Xiaoxing Ma. Neuro-symbolic data generation for math reasoning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b.

Hongcheng Liu, Yusheng Liao, Yutong Meng, and Yuhao Wang. Xiezhi: Chinese law large language model, 2023.

Bingfeng Luo, Yansong Feng, Jianbo Xu, Xiang Zhang, and Dongyan Zhao. Learning to predict charges for criminal cases with legal basis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2727–2736, 2017.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jian-Guang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, Yansong Tang, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. In *Proceedings of the 13th International Conference on Learning Representations*, 2025.

Ha-Thanh Nguyen. A brief report on lawgpt 1.0: A virtual legal assistant based on GPT-3. *CoRR*, abs/2302.05729, 2023.

OpenAI. Gpt-3.5 turbo. Technical report, 2023a.

OpenAI. Gpt-4. Technical report, 2023b.

Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. Large language models in medicine. *Nature Medicine*, 29:1930–1940, 2023.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023.

Yiquan Wu, Yuhang Liu, Yifei Liu, Ang Li, Siying Zhou, and Kun Kuang. Wisdom interrogatory. `https://github.com/zhihaiLLM/wisdomInterrogatory`, 2024.

Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Yansong Feng, Xianpei Han, Zhen Hu, Heng Wang, and Jianfeng Xu. CAIL2018: A large-scale legal dataset for judgment prediction. *CoRR*, abs/1807.02478, 2018.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. Fingpt: Open-source financial large language models. *CoRR*, abs/2306.06031, 2023.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. In *Proceedings of the 12th International Conference on Learning Representations*, 2024.

Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. A survey of knowledge-enhanced text generation. *ACM Computing Surveys*, 54(11s):227:1–227:38, 2022.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 400–410, 2024.

Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 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*, pp. 5218–5230, 2020.

A Implementation Details for KgDG

We implement the KgDG framework based on DeepSeek V3 model (DeepSeek-AI et al., 2024) and a knowledge based with 186,197 high-quality criminal legal documents and 152,452 civil legal documents. In our implementation, we call API of DeepSeek V3 model in parallel with a batch size of 16 and the generation process repeats until the number of generated data reaches 25K. The specific implementation details are as follows.

KgGen. We first use the Prompt for Generation of KgGen to select which type of legal document should be sampled to generate similar types of reasoning problems based on the example.

Prompt for Sampling of KgGen

给你一个JSON格式的法律领域的问题及其答案。其中，instruction字段指导如何回答问题，question字段中包含问题，answer字段中包含答案。
{JSON}
现在请你根据法律文书数据生成类似的问题，请问你需要什么类型的文书数据。可以选择的类型有：刑事法律文书、民事法律文书。请你选择一项并以JSON格式在type字段中返回。

Here, the example problem is provided in JSON format in '{JSON}'. The *Knowledge-Aware Sampler* first determines the appropriate legal document type based on the example problem. Then, it randomly samples a document from the knowledge base of that type and generates a new problem-answer pair, complete with extracted references and reasoning paths.

Prompt for Generation of KgGen

给你一个JSON格式的法律领域的问题及其答案。其中，instruction字段指导如何回答问题，question字段中包含问题，answer字段中包含答案。
{JSON}
请你以如下法律文书的内容为原型，按照相同的JSON格式和问题形式，在instruction不变的情况下，编造一个新问题与对应的答案。
请增加一个reasoning字段，此字段是一个字符串，表示得出答案的推理过程。
请增加一个reference字段，此字段是一个字典，Key为推理过程中涉及的法律法条，Value表示法律法条的具体内容。
请适当改写法律文书的内容，不要包含与答案无关的内容，不要直接复述法律文书的内容。
请修改问题与答案中的姓名、企业名称、地点等涉及隐私的内容。
answer字段的内容应该完全按照instruction中的对答案的格式要求给出。
{DOCS}

Here, the example problem is provided in JSON format in '{JSON}' and the sampled legal document is provided in '{DOCS}'.

KgFix. We first use the Prompt for Reference Modifier and Reasoning Corrector to correct the references and reasoning paths for each draft data.

Prompt for Reference Modifier

给你一个包含若干法条的JSON字典，此字段是一个字典，Key为推理过程中涉及的法律法条，Value表示法律法条的具体内容。
{JSON}
法条的内容可能存在问题，请你将Value修正为Key对应的正确法条内容，并以JSON格式返回，不要附加其他内容或说明。

Prompt for Reasoning Corrector

给你一个JSON格式的法律领域的问题及其答案。其中，instruction字段指导如何回答问题，question字段中包含问题，answer字段中包含答案，reference字段中包含法律法条的内容，reasoning包含推理过程。
{JSON}
当前问题的推理过程与答案可能存在问题，请根据问题内容、法律法条内容，改进当前的推理过程与答案。
如果此问题的推理过程与答案无需改进，请直接输出原始JSON格式内容，否则请修改reasoning字段和answer字段的内容后，直接输出JSON格式内容。不要附加其他内容或说明。

Here, the draft data is provided in JSON format in ‘{JSON}’.

DaVer. We first use the Prompt for Verification to verify the correctness of the generated question-answer pair as well as the consistency between the reasoning, reference and the answer.

Prompt for Verification

给你一个JSON格式的法律领域的问题及其答案。其中，instruction字段指导如何回答问题，question字段中包含问题，answer字段中包含答案，reference字段中包含法律法条的内容，reasoning包含推理过程。{JSON}
请你判断数据中的推理过程与答案是否正确，请以JSON格式返回你的判断结果。JSON格式数据中包含一个verify字段，取值为正确或错误，也包含一个message字段，表示你判断的理由。

Here, the draft data to be verified is provided in JSON format in ‘{JSON}’.