ClauseQA: Enhancing Customized Clause Extraction in Large Language Models via Instruction Following

Anonymous ACL submission

Abstract

Contract review is a critical and time-consuming task for lawyers, involving the identification of key clauses that may pose potential risks. However, previous methods trained on predefined taxonomies struggle to generalize to meet varying requirements. To address this limitation, we propose **ClauseQA**, a framework to adapt large language models (LLMs) to extract clauses by following instructions with customized clause descriptions. Additionally, we introduce an out-of-distribution setting for recognizing unseen clause categories, investigating how supervised fine-tuning (SFT) affects LLMs' generalization. Our experiments show that SFT significantly reduces hallucinations while making LLMs more cautious in providing positive answers, which can sometimes lead to lower recall. Furthermore, we observe that SFT tends to induce the original pre-training capability in decoder-only models like Llama3, whereas encoder-decoder models, such as Flan-T5, fit the SFT data more closely and thus show less robustness to distribution shifts. Finally, we discuss potential directions for future research. Our code and models will be released.

1 Introduction

Legal AI (Zhong et al., 2020) is attracting increasing attention for its potential to promote justice (Zhong et al., 2018; Chalkidis et al., 2019) and create economic value, such as assisting lawyers in reviewing contracts (Leivaditi et al., 2020; Hendrycks et al., 2021). Contract review aims to identify risks and revise clauses to protect their parties' interests. Reviewing entire contracts, which can span hundreds of pages, to find specific clauses is time-consuming. Significant time savings can be achieved by automatically locating and highlighting key clauses using AI models. However, lawyers' attention can change depending on the type of contracts, necessitating models that can adapt to customized instructions. This poses significant challenges for traditional AI models.

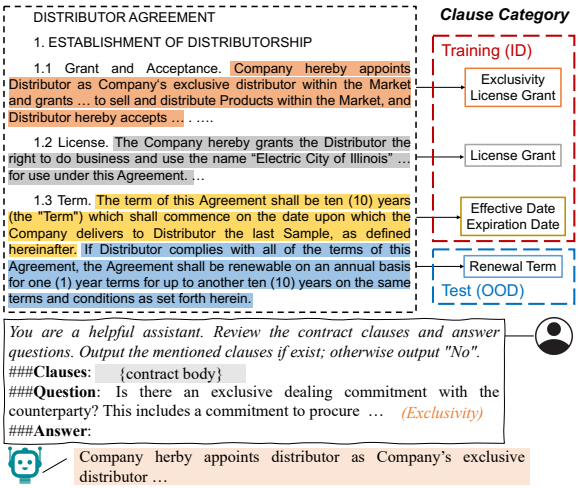

Figure 1: An example of a distributor agreement contract to illustrate the contract review task. Highlighted text spans denote different clause categories. Given a question describing the desired clause category (*Exclusivity*), LLMs are expected to generate the original clauses belonging to this category.

Previous works typically focus on extracting clause snippets for manual review based on a predefined taxonomy of clause categories (Leivaditi et al., 2020; Xu et al., 2022), namely *clause extraction*, as shown in Figure 1. These studies often formulate clause extraction as extractive QA (Rajpurkar et al., 2018), and train BERT-based models with supervised data. However, BERT-based models struggle with new questions for unseen clause categories, making them unsuitable for real-world applications. Recently, large language models (LLMs) (Achiam et al., 2023; Touvron et al., 2023; Jiang et al., 2023) have shown significant potential in industry, particularly in completing new tasks by following user instructions. Directly prompting LLMs has also been applied to legal tasks (Shui et al., 2023). Although flexible, the performance of directly prompting often falls short due to insufficient adaptation to the legal domain

(Niklaus et al., 2024). This necessitates supervised fine-tuning (SFT) (Wei et al., 2021; Ouyang et al., 2022) with task-specific instructions.

In this paper, we propose **ClauseQA** to tackle contract review by enabling LLMs generate desired clause snippets through following instructions in response to customized questions. We develop instruction data tailored to clause extraction for SFT and introduce an out-of-distribution (OOD) (Hendrycks et al., 2020b) setting to investigate how SFT affects model generalization. Specifically, we enrich prompt questions with detailed descriptions of clause characteristics, as shown in Figure 1, which facilitates the recall of relevant legal knowledge. Additionally, we incorporate negative samples to align models to appropriately reject questions when no applicable contract clause exists. To mirror real-world conditions, we divide clause categories into two distinct sets: an in-distribution (ID) set for training and an OOD set for testing. This division allows us to effectively evaluate the model's ability to generalize to unseen clause categories.

We conduct experiments using various LLMs, including Llama 3 (AI@Meta, 2024), Mistral (Jiang et al., 2023) and Flan-T5 (Chung et al., 2024). Our main findings are as follows: 1) SFT is important to induce pre-training capabilities, showing significant improvement over direct prompting. 2) While SFT helps reduce false positive answers (hallucinations), it also makes LLMs more cautious, leading to lower recall, sometimes. 3) Decoder-only and encoder-decoder LLMs benefit from SFT in different ways. SFT tends to induce the pre-training capabilities of the former (Zhou et al., 2024), whereas the latter tends to better fit the SFT data and thus show less robustness to distribution shifts.

We highlight two potential directions for future research. First, incorporating more legal knowledge in pre-training stage is critical for downstream task performance. Second, there is significant potential for developing advanced contract review systems, such as collaboration of multiple LLM-based agents (Qian et al., 2023; Zheng et al., 2024) akin to a team of expert legal advisors.

2 Methodology

We first formulate the clause extraction task and the training objective (Section 2.1), followed by the generation of SFT samples (Section 2.2). Finally, we introduce the OOD setting to test the models' generalization capabilities (Section 2.3).

2.1 Task Formulation

Clause extraction involves extract the original spans in contracts belonging to a given clause category. To enable efficient contract review, we follow the standard extractive QA formulation (Hendrycks et al., 2021). Let $\mathbf{c}$ represent a segment of a contract [1], and $\mathbf{q} \in \mathcal{Q}$ refers to a question in a clause category. The answers to question $\mathbf{q}$ are contract snippets, denoted as $\mathbf{x}^{\mathbf{q}}$. Then, the clause extraction task involves extracting target clauses for each question, defined as: $f : (\mathbf{c}, \mathbf{q}) \rightarrow \mathbf{x}^{\mathbf{q}}$. For each contract, we process each question $\mathbf{q}_i$ sequentially to extract all pertinent clauses.

Training Objective. Given a LLM with parameters θ, we fine-tune the model to directly maximize the conditional probability of the ground-truth contract snippets based on the prompts: $p_\theta(\mathbf{x}^{\mathbf{q}}|a(\mathbf{c}, \mathbf{q}))$, where the prompt function $a(\mathbf{c}, \mathbf{q})$ will be introduced later.

2.2 Training Samples of ClauseQA

We begin by introducing the design of the prompts used for training samples, followed by a discussion on the necessity of negative samples and their construction.

Prompt Design. The prompt template is shown in Figure 1, consisting of three parts: an instruction detailing the task and the desired output format, the context of the contract body, and a question describing the characteristics of target clauses. We employ descriptions rather than mere category names to help LLMs recall knowledge of legal terminologies, thereby enhancing their ability to generalize to unseen clause terms.

Reject Unknown Questions. If target clauses are missing in a contract, LLMs should properly reject the question and avoid providing with false positive answers, known as hallucination (Zhang et al., 2023; Huang et al., 2023). Therefore, we build *negative* samples where the ground answers of missing clauses are set to "No".

Sampling Negative Samples. Considering the sparsity of key clauses, we develop two strategies to construct negative samples. For each clause $\mathbf{q}$, we randomly sample segments where $\mathbf{q}$ is absent and ensure their number proportional to positive samples. Moreover, we emphasize distinguishing nuanced clause categories by questioning a contract segment containing $\mathbf{q}_i$ with a negative question $\mathbf{q}_j$.

[1] A contract is divided into segments due to GPU memory constraints. Details are introduced in Appendix B.1

Model	Macro				Micro			
	P	R	F1	IOU	P	R	F1	IOU
Direct Prompting								
Flan-T5-XL	18.34	51.97	23.38	13.69	14.44	46.13	21.27	12.01
Llama3-Chat	20.03	64.03	26.64	16.24	16.91	66.20	26.55	15.36
Supervised Fine-tuning (Fully)								
T5-Large	3.71	13.02	4.21	2.18	3.73	10.00	4.55	2.34
Flan-T5-Large	67.69	47.37	45.99	33.26	63.19	45.75	51.22	34.42
Flan-T5-XL	69.14	50.04	**49.75**	**36.30**	65.18	51.16	**56.52**	**39.42**
Supervised Fine-tuning (Lora)								
Llama3	68.05	43.88	44.10	31.73	64.58	44.85	52.20	35.48
Llama3-Chat	68.45	46.93	**48.76**	**35.45**	69.22	48.35	**56.85**	**39.74**
Mistral	67.03	45.96	45.80	33.33	63.84	46.25	51.91	35.10
Mistral-Chat	62.63	40.91	42.09	29.83	64.96	41.82	50.10	33.64

Table 1: The OOD performance of direct prompting and SFT . The best performances are highlighted in bold for model trained with fullly fine-tuning and PEFT separately.

2.3 OOD Setting

The OOD setting is to evaluate how LLMs generalize to recognize unseen clauses categories.

The clause categories are divided into two disjoint sets, denoted as $\mathcal{Q}^{ID}$ and $\mathcal{Q}^{OOD}$. Training and test contracts are denoted as $\mathcal{C}^{tr}$ and $\mathcal{C}^{te}$, respectively. The training data of SFT is constructed with training contracts and ID categories: $\{(\mathbf{c}, \mathbf{q}) | \mathbf{c} \in \mathcal{C}^{tr}, \mathbf{q} \in \mathcal{Q}^{ID}\}$, while the test data is constructed as $\{(\mathbf{c}, \mathbf{q}) | \mathbf{c} \in \mathcal{C}^{te}, \mathbf{q} \in \mathcal{Q}^{OOD}\}$, ensuring that both test contracts and clauses are unseen during training.

The above setting leads to the OOD performance. We also introduce the Full performance, where both ID and OOD categories are seen during training.

3 Experiment and Results

We first brief the experimental setup (Section 3.1), and introduce the findings to uncover the effects of SFT (Section 3.2-3.5).

3.1 Experimental Setup

Dataset and Pre-processing. We use the contract review dataset, CUAD (Hendrycks et al., 2021) that contains 510 commercial contracts and manual annotation of 41 clause categories. These contracts, averaging 7,861 words, are divided into segments based on paragraphs, with one-fifth containing key clauses. Dataset statistics and segmentation details are introduced in Appendix B.1

Dataset Splits. We follow the original CUAD division, with 408 training contracts and 102 test contracts. We specify the sizes of ID and OOD category sets as 29 and 12, respectively. We create three splits of the ID-OOD division using three random seeds, and performances are averaged across these three splits.

Models. We experiment on open-source LLMs, including Llama3 (8B) (AI@Meta, 2024), Mistral (7B) (Jiang et al., 2023) and both the large (800M) and xl (3B) versions of Flan-T5 (Chung et al., 2024). For Llama3 and Mistral, we employ Lora (Hu et al., 2021; Dettmers et al., 2024) adapters across all linear layers during training.

Metrics. Following Hendrycks et al. (2020a), we utilize word-level overlap of ground answers and generated outputs and calculate the precision, recall, F1 score and Intersection over Union (IOU) scores with Macro and Micro methods.

3.2 Overall Results

The performance on OOD clause categories are shown in Table 1. We highlight two findings.

SFT benefits task alignment, while prompting leads to under-estimation. We observe a significant improvement of SFT over prompting. This highlight the importance of task alignment (Zhou et al., 2024) to unlock the pre-training capabilities, and suggest the under-estimation of prompting.

General instruction tuning typically improve generalization. Chung et al. (2024) show that instruction tuning on multitasks yields generalization on new tasks. We observe enhanced generalization of the instruction-tuned version of Flan-T5 and Llama3. However, this situation is opposite for Mistral. The degradation of Mistral-Chat mainly results from one split. We speculate that the instruction tuning data of Mistral includes few contract knowledge and distracts the base LM.

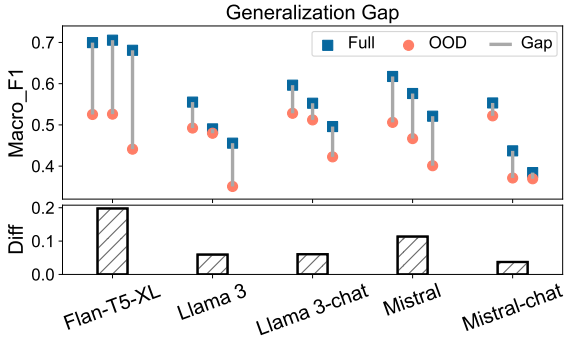

Figure 2: Generalization gap of different LLMs. The upper part shows performance of three splits, and the difference between "Full" and "OOD" performance is plotted at bottom.

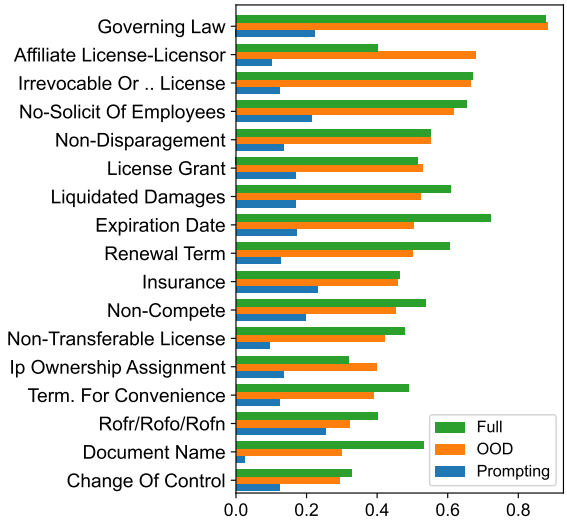

Figure 3: Performance across clause categories. The Macro IOU of Full, OOD and Prompting performance of Llama3-Chat is visualized.

3.3 Generalization Gap

We measure the robustness to distribution shifts by the difference between Full and OOD performance (generalization gap), as shown in Figure 2.

The result demonstrates a noticeable decrease of the gap between larger LMs and smaller ones, indicating that LMs with more parameters tend to be more robust. However, the result of two Mistral models suggests that capable models may suffer more drop to distribution shifts.

Figure 3 shows performance across clause categories. Comparing Full and SFT performance, we argue that **LLMs after SFT can generalize well to unseen clauses**.

3.4 SFT Leads to More Cautious Models

One interesting finding in Figure 3 is that the Full performance is even worse than OOD performance on some categories. It is **counter-intuitive that models' capabilities in these categories degenerate after seeing them during SFT**.

Model	**Flan-T5**	**Llama3**	**Mistral**
Corr	0.37	0.83 (0.88)	0.71 (0.88)

Table 2: Correlation between Full and OOD performance across clause categories. The values in parentheses denote the correlation for Chat models.

A close check on these categories reveal that there is a large drop in recall while the precision usually remains similar or even improves. Similar phenomenon is also observed in Table 1. Comparing prompting and SFT, we find that SFT enhance the holistic precision and decrease recall to some extend. It suggests that SFT reduces hallucination, while the side effect is a more cautious model in providing positive responses.

We speculate two reasons for cautious models: first, the existence of nuanced clause categories, e.g., Cap on Liability and Uncapped Liability; second, the existence of negative samples influencing the output distribution.

3.5 SFT Induces Pre-training Ability or Introduces New Knowledge?

To uncover the effect of SFT, we calculate the correlation between Full and OOD performance across clause categories in Table 2. The intuition is that LLMs acquire different knowledge levels of these clause categories during pre-training, which can be reflected by the OOD performance.

We find that decoder-only LLMs demonstrate a high correlation between performance before and after seeing specific clauses during SFT, while encoder-decoder LLMs an indistinct correlation. This implies that the pre-training capabilities of decoder-only LMs tend to be induced by SFT, while encoder-decoder LMs tend to fit the SFT data and are less robust to distribution shifts.

4 Conclusion

We proposed ClauseQA, a framework for adapting LLMs to extract desired clauses in contracts based on instructions with customized descriptions of clause characteristics. This framework is practical in real-world applications, enabling LLMs to generalize to unseen clause categories. We conducted an in-depth analysis to reveal the side effects of SFT, observed as producing "cautious" models, and the different behaviors of decoder-only and encoder-decoder LLMs during SFT.

Limitation

We discuss three key limitations related to the fine-tuning technique, objective, and dataset used in our study.

First, our models with 7B parameters or more are fine-tuned with parameter efficient technique, Lora. This may result in sub-optimal performance. Due to the limitation of GPU memory size (NVIDIA A5000 with 24GB memory), we have to compromise performance with GPU memory usage. Future work will compare the performance of Lora and fully fine-tuning.

Second, the training objective of SFT is to only maximize the probability of ground-truth answers. Alternative fine-tuning methods, such as Direct Preference Optimization (DPO), can be utilized to better train the model.

Third, our experiments are limited to a single dataset comprising contracts from the US Securities and Exchange Commission (SEC). Due to the high cost of labeling contract clauses, there are not many available contract review datasets with high quality. We will incorporate more datasets in future work.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

AI@Meta. 2024. Llama 3 model card.

Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019. Neural legal judgment prediction in english. *arXiv preprint arXiv:1906.02059*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Neel Guha, Julian Nyarko, Daniel Ho, Christopher Ré, Adam Chilton, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel Rockmore, Diego Zambrano, et al. 2024. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Advances in Neural Information Processing Systems*, 36.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020a. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. 2021. Cuad: An expert-annotated nlp dataset for legal contract review. *arXiv preprint arXiv:2103.06268*.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020b. Pretrained transformers improve out-of-distribution robustness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2744–2751.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Spyretta Leivaditi, Julien Rossi, and Evangelos Kanoulas. 2020. A benchmark for lease contract review. *arXiv preprint arXiv:2010.10386*.

Joel Niklaus, Lucia Zheng, Arya D McCarthy, Christopher Hahn, Brian M Rosen, Peter Henderson, Daniel E Ho, Garrett Honke, Percy Liang, and Christopher Manning. 2024. Flawn-t5: An empirical examination of effective instruction-tuning data mixtures for legal reasoning. *arXiv preprint arXiv:2404.02127*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Chen Qian, Xin Cong, Cheng Yang, Weize Chen, Yusheng Su, Juyuan Xu, Zhiyuan Liu, and Maosong Sun. 2023. Communicative agents for software development. *arXiv preprint arXiv:2307.07924*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Ruihao Shui, Yixin Cao, Xiang Wang, and Tat-Seng Chua. 2023. A comprehensive evaluation of large language models on legal judgment prediction. *arXiv preprint arXiv:2310.11761*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Don Tuggener, Pius Von Däniken, Thomas Peetz, and Mark Cieliebak. 2020. Ledgar: A large-scale multi-label corpus for text classification of legal provisions in contracts. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1235–1241.

Steven H Wang, Antoine Scardigli, Leonard Tang, Wei Chen, Dimitry Levkin, Anya Chen, Spencer Ball, Thomas Woodside, Oliver Zhang, and Dan Hendrycks. 2023. Maud: An expert-annotated legal nlp dataset for merger agreement understanding. *arXiv preprint arXiv:2301.00876*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Weiwen Xu, Yang Deng, Wenqiang Lei, Wenlong Zhao, Tat-Seng Chua, and Wai Lam. 2022. Conreader: Exploring implicit relations in contracts for contract clause extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2581–2594.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Jingnan Zheng, Han Wang, An Zhang, Tai D Nguyen, Jun Sun, and Tat-Seng Chua. 2024. Ali-agent: Assessing llms' alignment with human values via agent-based evaluation. *arXiv preprint arXiv:2405.14125*.

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.

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. *arXiv preprint arXiv:2004.12158*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

A Related Work

A.1 LLMs for Legal Domain

Recently, there are increasing work focusing on applying LLMs on legal tasks, evaluating their legal capabilities, and building legal domain resources.

For evaluation close-source leading LLMs, GPT-4 (Achiam et al., 2023) is reported to pass the Uniform Bar Exam (UBE) with a 90 percentile score. However, the benchmark of UBE is argued to be too general. Therefore, Shui et al. (2023) propose a new benchmark to evaluate LLMs' capacity on legal judgment prediction with retrieval augmented prompts. Guha et al. (2024) propose a large scale benchmark, namely LegalBench, that encompass various legal tasks. LegalBench also includes the dataset CUAD (Hendrycks et al., 2021) used in this paper. However, CUAD is simplified as answering yes or no questions in LegalBench, while we keep the original settings to extract key clauses from whole contracts.

Recently, Niklaus et al. (2024) build a large scale instruction tuning dataset for legal reasoning, and investigate the effects of continued pre-training and instruction tuning.

A.2 Legal Contract Review

Legal contracts, especially commercial contracts, are receiving attention due to their importance in business activities. Most of the datasets are built from the corpus provided by the US Security and Exchange Commission.

Tuggener et al. (2020) focus on classify clauses (provisions) into their paragraph headings, and the label taxonomy is built based on heuristics. The following work begins to adopt expert defined clause categories, such as red flags (Leivaditi et al., 2020) and CUAD (Hendrycks et al., 2021). Contract reading comprehension datasets are also created for specific contract types, such as merger agreements (Wang et al., 2023).

These datasets are usually published with naive baseline methods, such as BERT-based models. Some studies delve into the structure and semantic relations in contracts to enhance performance(Xu et al., 2022).

B Experiment Settings

B.1 Data Processing

Statistics. Table 3 presents dataset statistics on contract length and key clauses density. The first two

rows indicate that contracts typically contains substantial paragraphs. However, only about one-fifth of these paragraphs contain key clauses necessitating thorough legal review (Row 3). Moreover, the review of contracts across various types requires emphases on distinct clause categories. On average, a contract contains approximately 13 categories of key clauses, with each category appearing in 31.5% of contracts (Row 4-5).

The statistics underscore the importance of leveraging LLMs to assist lawyers in this needle-in-a-haystack task, as well as developing robust models capable of managing emerging contract categories.

Average no. of para. per doc.	65.5
Average no. of words per para.	120.0
Percent of para. w/ key clauses	19.3
Average no. of clause types per doc.	12.9
Average ratio of clause occurrence	31.5

Table 3: Statistics of the CUAD dataset about the number of paragraphs and clauses to show the sparsity of key clauses.

Contract Segmentation. Different from Hendrycks et al. (2021), who utilize a sliding window approach to segment contracts, we first divide contracts into segments based on paragraphs, treating each paragraph as a separate segment [2]. This method often results in many short paragraphs, typically chapter headings. To address this, we iteratively merge each short paragraph with the subsequent one until the combined length exceeds a threshold of 300 characters. For longer paragraphs, we further split them into consecutive segments, each with a window size of 512 tokens.

B.2 Evaluation Metrics

Following (Hendrycks et al., 2021), we utilize the word-level overlap of ground-truth answer and LLM generated predictions to evaluate the performance of clause extraction. We introduce four used metrics, namely precision, recall, F1 score and Intersection over Union (IOU) below.

Let $Gold$ denote the number of words in ground-truth answers and $Pred$ the number of words in generated text [3]. The correctly predicted clause snippets contain $Join$ words. The four metrics are calculated as:

$$P = \frac{Join}{Pred}; \quad R = \frac{Join}{Gold};$$
$$F1 = \frac{2 * P * R}{P + R} = \frac{2 * Join}{Pred + Gold}; \quad (1)$$
$$IOU = \frac{Join}{Pred + Gold - Join}$$

The $Micro$ metrics are calculated by aggregating word counts across all clause categories and contracts. $Macro$ metrics are derived by first aggregating counts for each clause category across all contracts , and then averaging these metrics across all clause categories [4].

B.3 Training Details

We introduce the training hyper-parameters, hardware and training cost here.

We use learning rate of $1e-5$ for decoder-only LLMs and $1e-4$ for encoder-decoder LLMs. The weight decay is set to 0.0. We use two NVIDIA A5000 gpus to train Flan-T5-XL, Llama3 and Mistral. We train all models for 5 epochs and it costs 5 to 7 hours for one running.

[2] The terms 'paragraph' and 'segment' are used interchangeably below.

[3] $Pred$ is set to 0 if a model refuses answering with "No"

[4] The aggregation methods are different from (Hendrycks et al., 2021)