Team UTSA-NLP at SemEval 2024 Task 5: Prompt Ensembling for Argument Reasoning in Civil Procedures with GPT4

Dan Schumacher[1] and **Anthony Rios**[2]
[1]MS Data Analytics, College of Business
[2]Department of Information Systems and Cyber Security
The University of Texas at San Antonio
dan.schumacher@my.utsa.edu, anthony.rios@utsa.edu

Abstract

In this paper, we present our system for the SemEval Task 5, *The Legal Argument Reasoning Task in Civil Procedure* Challenge. Legal argument reasoning is an essential skill that all law students must master. Moreover, it is important to develop natural language processing solutions that can reason about a question given terse domain-specific contextual information. Our system explores a prompt-based solution using GPT4 to reason over legal arguments. We also evaluate an ensemble of prompting strategies, including chain-of-thought reasoning and in-context learning. Overall, our system results in a Macro F1 of .8095 on the validation dataset and .7315 (5th out of 21 teams) on the final test set. Code for this project is available at https://github.com/danschumac1/CivilPromptReasoningGPT4.

1 Introduction

Mastering the reasoning behind legal arguments is a fundamental skill required of all law students. In this study, we develop a novel approach for SemEval Task 5, The Legal Argument Reasoning Task in Civil Procedure Challenge (Held and Habernal, 2024). The SemEval Task released a dataset that was scraped from *The Glannon Guide To Civil Procedure*, a textbook designed for law students. Specifically, given case law, a question, and a potential answer to that question, students must be able to reason over the contextual information (case law) to determine if the question is correct or not.

There has been substantial research in developing NLP-based reasoning systems (Guha et al., 2024; Bongard et al., 2022; Chalkidis, 2023; Blair-Stanek et al., 2023; Kuppa et al., 2023; Yu et al., 2022). The methods can be categorized into two major frameworks: fine-tuning and large language model-based (LLM) solutions. For fine-tuning approaches, Bongard et al. (2022) introduced an approach that fine-tunes LegalBERT (Chalkidis et al., 2020) and developed several methods for handling long text that does not fit within the token limitations of LegalBERT. For LLM solutions, Chalkidis (2023) explored the use of ChatGPT for solving legal exams. Guha et al. (2024) introduced a more comprehensive evaluation benchmark geared to large language models consisting of 162 tasks. Many of their experiments show that GPT4 is one of the top performing approaches for legal reasoning across all language models, while Flan-T5-XXL is the best open-source option. Although showing substantial generalization is important, developing specific prompting strategies for different reasoning tasks can substantially improve performance. Hence, this paper adds to existing literature on the exploration of prompting approaches in the legal domain.

For this work, we adapt several prompting-based strategies to develop an LLM-based solution for the shared task. Specifically, we combine a retrieval system with in-context learning and chain-of-thought reasoning. There are several studies showcasing the utility of in-context learning (Liu et al., 2022a; Lu et al., 2022) and chain-of-thought reasoning (Wei et al., 2022; Sun et al., 2023; Yao et al., 2024). Moreover, there is work using both human-curated and machine-generated reasons. In this work, we focus on human-generated reasons for the training data, and machine-generated examples are only used at test times when human expert annotations are not provided. Furthermore, rather than providing a step-by-step reasoning approach, which may not make sense in this context, our approach is more similar to single-step rationales (Brinner and Zarrieß, 2023; Yasunaga et al., 2023), which simply provides a single step of reasoning for why an answer is correct or incorrect.

In summary, this paper makes the following contributions to our solution for the SemEval 2024 Task 5 shared task:

- We evaluate prompting strategies using GPT4

that combine several popular ideas, including in-context learning and chain-of-thought reasoning.

- In-context learning can be sensitive to the actual choice of examples, particularly when only a few examples are provided. Hence, we also explore an ensemble of prompt-based predictions to improve overall performance.

- Finally, we provide a unique error analysis where we found limitations and common error types generated by GPT4 using our prompting strategies. For example, when a part of an answer candidate is correct, but the reasoning is wrong, GPT4 is likely to generate a false positive (i.e., predict it is correct instead of incorrect).

2 RELATED WORK

Overall, there are three major areas of legal NLP: legal question-answering (Khazaeli et al., 2021; Kien et al., 2020; Ryu et al., 2023; Martinez-Gil, 2023; Wang et al., 2023), judgment prediction (Masala et al., 2021; Valvoda et al., 2023; Juan et al., 2023), and corpus mining (e.g., summarization, text classification, information extraction, and retrieval-related research) (Poudyal et al., 2020; Li et al., 2022; Vihikan et al., 2021; Zhang et al., 2023; Limsopatham, 2021; de Andrade and Becker, 2023). There has also been some broad methodology work that is aimed at working on various legal tasks in general (e.g., LegalBERT (Chalkidis et al., 2020)).

The SemEval task is most similar to question-answering related research. In the domain of legal question-answering, recent research efforts are focused on creating new systems, developing evaluation criteria, and compiling datasets, considering the significant variation across different legal fields. Khazaeli et al. (2021) introduced a commercial question-answering system for legal inquiries, leveraging information retrieval techniques, sparse vector search, embeddings, and a BERT-based reranking system, trained on both general and legal domain data. Ryu et al. (2023) developed a novel evaluation method for LLM-generated texts that assess their validity using retrieval-augmented generation, showing improved alignment with legal experts' assessments and effectiveness in identifying factual errors. Wang et al. (2023) created the Merger Agreement Understanding Dataset (MAUD), a unique, expert-annotated dataset for legal text reading comprehension, highlighting promising model performance and the need for further improvement in understanding complex legal documents.

From a methodological point-of-view instead of a task-oriented view, recent research efforts have concentrated on the advancement of NLP-based reasoning systems, with a particular focus on applications within the legal domain (Guha et al., 2024; Bongard et al., 2022; Chalkidis, 2023; Blair-Stanek et al., 2023; Kuppa et al., 2023; Yu et al., 2022). These efforts can be broadly classified into two distinct methodologies: fine-tuning approaches and those leveraging large language models (LLMs). Within the fine-tuning paradigm, Bongard et al. (2022) have proposed modifications to LegalBERT (Chalkidis et al., 2020) aimed at enhancing its ability to process texts that exceed the model's inherent token limitations. This approach is representative of a broader trend towards tailoring pre-existing models to better suit specific textual analysis tasks in the legal sector. Kien et al. (2020) developed a retrieval-based model employing neural attentive text representation with convolutional neural networks and attention mechanisms for accurately matching legal questions to relevant articles, demonstrating superior performance on a Vietnamese legal question dataset.

Conversely, the exploration of LLMs has also been wide covering new general approaches for legal question answering and reasoning to new datasets and benchmarks. Yu et al. (2023) investigated the impact of chain-of-thought prompts and fine-tuning methods on legal reasoning tasks, specifically the COLIEE entailment task, and found that prompts based on legal reasoning techniques and few-shot learning with clustered training data significantly enhance performance. Chalkidis (2023) demonstrates the potential of utilizing models like ChatGPT for complex reasoning tasks, such as solving legal examination questions. Building on this, Guha et al. (2024) have introduced a comprehensive evaluation framework designed specifically for assessing the capabilities of LLMs across a suite of 162 tasks. This benchmark aims to provide a more nuanced understanding of the strengths and limitations of LLMs in the context of legal reasoning. Additionally, while employed within a distinct domain from legal reasoning, a comparable methodology in prompt engineering—encompassing chain of thought prompting

and in-context prompting—is illustrated in Liu et al.'s recent work (Liu et al., 2022b). Overall, compared to the prior work that developed methods and datasets for generating answers to questions, the SemEval task focuses on understanding whether a provided answer candidate is valid given a specific context.

3 METHOD

We provide a high-level overview of our approach in Figure 1. Overall, we explore three major different prompting approaches: Zero-shot prompting, few-shot prompting, and few-shot prompting with chain-of-thought-like reasoning. Moreover, we explore an ensemble of multiple approaches. The methods are described in the following subsections.

3.1 Few-Shot Retrieval Augmented Chain-of-Thought Prompting

We refer to our approach as "Few-Shot & CoT & RAG." Each example within the dataset contains an Introduction, Question, Answer Candidate, Analysis, and Label. For new examples at test time, we only have access to the Introduction, Question, and Answer Candidate. The Introduction consists of a general background about a legal case. The Question is about the case, and the Answer Candidate is an answer to the Question. It is important to note that the Question could be in question form, where the answer directly answers what is asked. However, it may function as a fill-in-the-blank exercise, where the question presents an incomplete statement that the Answer Candidate is expected to complete. The Analysis, which is only provided in the training and development datasets, is a detailed expert-defined explanation for why the Answer Candidate is or is not valid. The Label is a TRUE or FALSE value, where TRUE means that the Answer Candidate correctly addresses the question given the provided context. Likewise, FALSE means that the Answer Candidate is incorrect.

As shown in Figure 1, our prompting strategy contains three main components, a system prompt, the in-context examples, and the final test instance we will classify as TRUE or FALSE. The system prompt describes what the large language model (LLM) should do. In the Figure, it is shown that we also added explicit information to limit the model from generating non-relevant information.

The in-context examples are provided to the LLM before the final text instance. Intuitively, the goal is to provide some examples of the task we are accomplishing to better ground the LLM to make better-generated responses. This is the Retrieval Augmented aspect of our system (i.e., RAG). While any random examples could be provided, we search for the most relevant examples for each test instance. Formally, given an input instance x_i consisting of a concatenated Introduction, Question, and Answer triplet w, we retrieve the most similar examples $\{x_1, \ldots, x_{\mathcal{N}(x_i)}\}$, where $\mathcal{N}(x_i)$ are the k most similar examples to x_i. Each question-answer pair is embedded using LegalBERT. The in-context examples all come from the provided training dataset. Once retrieved, all of the relevant information for the retrieved examples are used in the prompt (i.e., the Introduction, Question, Answer Candidate, Analysis, and Label). Figure 1 shows an example with two in-context examples.

Finally, the last component of the prompt is the text example. Basically, for every example we wish to make a prediction for, we pass the Introduction, Question, and Answer Candidate. The model will first generate the Analysis for that example, then it will generate the Label.

3.2 Ensemble

Besides the method described in the previous section, we also explored prompting variants to create an ensemble. We describe each of the additional methods below (besides the Few-SHOT & CoT & RAG method described in the previous subsection).

Zero-shot. This approach does not use any in-context examples. We provide the system prompt and the test example to the model directly to get the final prediction.

Zero-shot & CoT. This method builds on the Zero-Shot approach by adding the CoT aspect to the test example. Note that this is not available at test time, so the model generates an Analysis section without previous examples.

Few-Shot. The few-shot approach will use multiple in-context examples. However, unlike our main method explained in the previous section, it uses the same in-context examples for all test cases. The examples were chosen in an ad-hoc manner with an emphasis on relatively short examples to limit the number of tokens to reduce costs.

Few-Shot & CoT. This builds on the Few-Shot method, where ad-hoc in-context examples are still

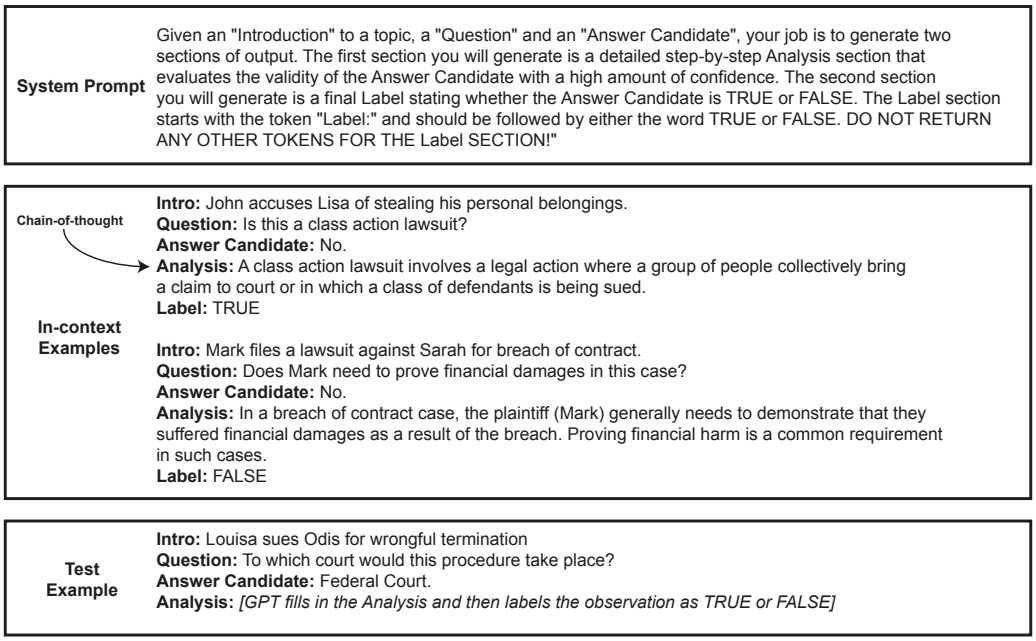

Figure 1: Overview of our prompting strategy.

used, but we also provide the CoT reasoning (Analysis) section to the in-context examples. The GPT4 model will generate the Analysis section for each test example.

Few-Shot & RAG. This also builds on the Few-Shot method, but instead of using ad-hoc examples, it uses LegalBERT and cosine similarity to find relevant in-context examples for each test case (as explained in the previous section).

Overall, there are a total of 4 models in our ensemble. Few-Shot and Few-Shot & RAG were not used, but we evaluated them. The models in the ensemble were chosen by checking combinations on the validation dataset. To make a prediction, we use voting with a threshold (i.e., where the votes are processed to generate the proportion of TRUE values).

3.3 Model Details

For all of our prompts, we use GPT-4-1106-preview with a temperature of .7. The similarity metric used for finding relevant in-context examples is cosine similarity. Moreover, we choose a total of 2 in-context examples, consisting of 1 TRUE example and 1 FALSE example. We searched for the best threshold after voting by calculating the proportion of TRUE vs. FALSE predictions, ultimately choosing .5 as the threshold.

There were many cases where GPT4 does not return a final label in an easy-to-process fashion (i.e., it does not end with a TRUE or FALSE). We explored multiple approaches to parse the answer. Our ultimate strategy involves re-submitting examples to GPT-4 that initially failed to produce a valid label, explicitly indicating the absence of the label, and prompting the model to generate the correct information.

Method	Macro F1	Acc.
Baselines		
RoBERTa	.5128	.2286
Legal-BERT	.5575	.2941
Zero-shot	.6681	.7857
Zero-Shot & COT	.7162	.7500
Few-shot	.6935	.7738
Few-shot & COT	.6762	.7262
Few-shot & RAG	.6898	.7500
Few-Shot & COT & RAG (Ours)	.7306	.7857
Ensemble (Ours)	.8095	.8571

Table 1: Validation dataset results

4 RESULTS

In this section, we present our own results on the validation dataset as well as the final results in the competition.

Baselines In our experiments, using the validation data, we compare our approach (Few-SHOT & CoT & RAG) with the other approaches used in the Ensemble. Moreover, we compare our system to both RoBERTa (Liu et al., 2019) and Legal-BERT. However, because both RoBERTa and Legal-BERT are limited to 512 tokens, we split the Introduction into b pieces. b is calculated by subtracting the number

of words in the question and answer from 512. We halve that result to accommodate the fact that the number of tokens typically exceeds the number of words. Next, we divide the number of words in the Introduction by the previously calculated number to obtain the total number of windows. Finally, all of the tokens in the Introduction are evenly split into the windows. Each piece of the introduction is appended to the Question and Answer pair independently to generate multiple predictions for each instance. We then use voting to make a final prediction for the entire sequence. This method is similar to what was explored in Bongard et al. (2022).

Validation Results. The validation performances are shown in Table 1. We observe that the fine-tuned methods (e.g., RoBERTa), perform less effectively compared to all methods utilizing GPT-4. Between the fine-tuned methods, we find that Legal-BERT outperforms RoBERTa, which is expected given Legal-BERT was fine-tuned on relevant corpora.

Next, between GPT4 methods (not including the ensemble), we find that performance varies substantially between .6681 and .7162 for Macro F1. All methods outperform Zero-Shot. However, Zero-Shot & COT achieved the best performance across all baseline methods for Macro-F1. When we compare the baseline approaches to our method (Few-Shot & COT & RAG), we find that the method outperforms all variations. From an ablation standpoint, removing RAG has the biggest performance drop (.7306 vs. .6762). Interestingly, we find that removing CoT has the second largest drop in performance (.7306 vs. .6935), and removing the in-context examples has the smallest drop in performance (.7306 vs. .7162). Before the study, we expected removing the in-context examples would result in the largest performance drop.

Competition Results. In Table 2, we report the final results of the competition, achieving a Macro F1 of .7315. But, why do we see such a large performance drop between the competition and validation results (.7315 vs. .8095)? We hypothesize two major reasons. First, we realized that the validation dataset contains many Introduction-Question pairs identical to the ones in the training dataset. Despite the Answer Candidates differing across the two datasets, the substantial overlap in Introduction-Question pairs may lead to an overestimation of our model's performance on the validation dataset,

#	User	Macro F1	Acc.
1	zhaoxf4	.8231	.8673
2	irene.benedetto	.7747	.8265
3	kbkrumov	.7728	.8367
4	qiaoxiaosong	.7644	.8163
5	**UTSA-NLP**	**.7315**	**.7959**
6	kubapok	.6971	.7857
7	samyak	.6599	.7449
8	hrandria	.6327	.6939
9	Yuan_Lu	.6000	.6327
10	PengShi	.5910	.6735
11	msiino	.5597	.5714
12	Hwan_Chang	.5556	.5918
13	kriti7	.5511	.6020
14	woody	.5510	.6633
15	odysseas_aueb	.5143	.6122
16	Manvith_Prabhu	.4966	.6224
17	lhoorie	.4957	.5000
18	yms	.4827	.7245
19	U_201060	.4503	.6633
20	langml	.4375	.4490
21	lena.held	.4269	.7449

Table 2: Final Competition Results. Our submission is in **bold** font.

	predFalse	**predTrue**
actFalse	57	9
actTrue	3	15

Table 3: Confusion matrix for the validation dataset. actFalse and actTrue stand for actual True and False values, respectively. predFalse and predTrue stand for predicted False and predicted True.

rendering it potentially too optimistic when applied to entirely new data. Second, we spent time over-optimizing the ensemble on the validation dataset causing overfitting issues (e.g., checking thresholds, model combinations, and more). By generating a better validation split, we may have seen better generalization.

Error Analysis Our method resulted in twelve mistakes on the validation dataset: ten false positives and two false negatives. The confusion matrix is shown in Figure 3. In Table 4 (See Appendix), we categorize each false positive into one of four categories: "Incorrect reasoning," "Shared the same introduction and question pair," "lots of similar language", and "Other." With only two false negatives, there are few useful patterns to understand among the errors. Hence, we only have a general FN category. However, from the false positives, we make two major findings which we describe below.

The first was when the answer candidate had the correct answer but *incorrect reasoning*. Here is a toy example demonstrating this pattern:

> **Introduction**: Carlos enjoys riding his skateboard in the skate park. Unfortunately, during one of his rides, he fell and split his head open.
>
> **Question**: Should Carlos go to the hospital?
>
> **Answer Candidate**: Yes, Carlos should go to the hospital because he likes to kick-flip so much.
>
> **Analysis**: While it is correct that Carlos should go to the hospital for medical attention after splitting his head open, the reasoning provided in the answer candidate is flawed. The decision to seek medical help should be based on the severity of the injury and the need for professional medical treatment, not on Carlos's enjoyment of skateboard stunts.

Intuitively, part of the Answer Candidate is correct, i.e., Carlos *should* go to the hospital, yet the reasoning that states why he should go to the hospital is wrong.

The second point is that three of our ten false positives all *shared the same introduction and question pair.* The introduction contained more extraneous information than usual and was 128 words longer than the average. In these instances, our model would analyze the answer candidate as correct but without taking into account the particular case that was asked about. Below is a toy example:

> **Introduction**: My dog Louisa loves to learn new tricks, go for walks, eat her dinner, then sleep through the night
>
> **Question**: What does Louisa like to do *after dinner*?
>
> **Answer Candidates**:
>
> 1. Learn new tricks (*wrong*)
> 2. Go for walks (*wrong*)
> 3. Eat her dinner (*wrong*)
> 4. Sleep through the night (*correct*)

In the example, three of the answers are incorrect, while "Sleep through the night" is the correct example. When the Introduction is long, the actual context of the question may be ignored, which can be interpreted as a needle in a haystack issue. Developing better systems that provide direct "attention" to relevant information may improve performance.

Finally, a third point that we believe caused our model to predict incorrect answer candidates is *similar language* in both the introduction and question compared to the answer candidate.

> **Introduction:** While making eggplant Parmesan for the first time in a buttery dutch over Paige burned herself on the stove
>
> **Question:** How does Paige remember this incident.
>
> **Answer Candidates:** She remembers making eggplant Parmesan for the first time in a buttery dutch oven fondly.

5 CONCLUSION

In this paper, we described our approach for 2024 SemEval Task 4, The Legal Argument Reasoning Task in Civil Procedures. Specifically, we introduced a GPT4 prompting-based strategy that achieved 5th place in the competition out of 21 participants. Overall, we find that combining in-context learning, where we use a retrieval-based approach to find relevant examples, as well as in-context learning improves model performance. Based on our experiments, there are three natural areas for future research. First, the actual Analysis section used for chain-of-thought reasoning does not match traditional methods which use step-by-step reasoning. Hence, a logical next extension is to reword (potentially with GPT4) the Analysis section to provide a step-by-step explanation for an answer. Second, this work was limited to 2 in-context examples to limit API costs and allow us to test other models in our initial experiments. However, extending that to 10 or more examples can potentially improve performance. Third, the current approach relies on a closed-source model (GPT4). Exploring open-source models, particularly smaller open-source models such as T5 (Chung et al., 2022) and LLama2 (Touvron et al., 2023), is important to better understand the impact of pretraining data on performance.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation (NSF) under Grant No. 2145357.

References

Andrew Blair-Stanek, Nils Holzenberger, and Benjamin Van Durme. 2023. Can gpt-3 perform statutory reasoning? *arXiv preprint arXiv:2302.06100*.

Leonard Bongard, Lena Held, and Ivan Habernal. 2022. The legal argument reasoning task in civil procedure. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 194–207.

Marc Brinner and Sina Zarrieß. 2023. Model interpretability and rationale extraction by input mask optimization. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13722–13744.

Ilias Chalkidis. 2023. Chatgpt may pass the bar exam soon, but has a long way to go for the lexglue benchmark. *arXiv preprint arXiv:2304.12202*.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. Legal-bert: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Leonardo de Andrade and Karin Becker. 2023. Bb25hlegalsum: Leveraging bm25 and bert-based clustering for the summarization of legal documents. In *Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing*, pages 255–263.

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.

Lena Held and Ivan Habernal. 2024. SemEval-2024 Task 5: Argument Reasoning in Civil Procedure. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, Mexico City, Mexico. Association for Computational Linguistics.

Yining Juan, Chung-Chi Chen, Hsin-Hsi Chen, and Daw-Wei Wang. 2023. Custodiai: A system for predicting child custody outcomes. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 10–16.

Soha Khazaeli, Janardhana Punuru, Chad Morris, Sanjay Sharma, Bert Staub, Michael Cole, Sunny Chiu-Webster, and Dhruv Sakalley. 2021. A free format legal question answering system. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 107–113.

Phi Manh Kien, Ha-Thanh Nguyen, Ngo Xuan Bach, Vu Tran, Minh Le Nguyen, and Tu Minh Phuong. 2020. Answering legal questions by learning neural attentive text representation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 988–998.

Aditya Kuppa, Nikon Rasumov-Rahe, and Marc Voses. 2023. Chain of reference prompting helps llm to think like a lawyer. In *Generative AI+ Law Workshop*.

Xiang Li, Jiaxun Gao, Diana Inkpen, and Wolfgang Alschner. 2022. Detecting relevant differences between similar legal texts. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 256–264.

Nut Limsopatham. 2021. Effectively leveraging bert for legal document classification. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 210–216.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. 2022a. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114.

Yilun Liu, Shimin Tao, Weibin Meng, Jingyu Wang, Wenbing Ma, Yuhang Chen, Yanqing Zhao, Hao Yang, and Yanfei Jiang. 2022b. Interpretable online log analysis using large language models with prompt strategies. In *Proceedings of the 3rd Wordplay: When Language Meets Games Workshop (Wordplay 2022)*, Seattle, United States. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. 2022. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. In *The Eleventh International Conference on Learning Representations*.

Jorge Martinez-Gil. 2023. A survey on legal question–answering systems. *Computer Science Review*, 48:100552.

Mihai Masala, Radu Cristian Alexandru Iacob, Ana Sabina Uban, Marina Cidota, Horia Velicu, Traian Rebedea, and Marius Popescu. 2021. jurbert: A romanian bert model for legal judgement prediction. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 86–94.

Prakash Poudyal, Jaromír Šavelka, Aagje Ieven, Marie Francine Moens, Teresa Goncalves, and Paulo Quaresma. 2020. Echr: Legal corpus for argument mining. In *Proceedings of the 7th Workshop on Argument Mining*, pages 67–75.

Cheol Ryu, Seolhwa Lee, Subeen Pang, Chanyeol Choi, Hojun Choi, Myeonggee Min, and Jy-Yong Sohn. 2023. Retrieval-based evaluation for llms: A case study in korean legal qa. In *Proceedings of the Natural Legal Language Processing Workshop 2023*, pages 132–137.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023. Text classification via large language models. *arXiv preprint arXiv:2305.08377*.

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*.

Josef Valvoda, Ryan Cotterell, and Simone Teufel. 2023. On the role of negative precedent in legal outcome prediction. *Transactions of the Association for Computational Linguistics*, 11:34–48.

Wayan Oger Vihikan, Meladel Mistica, Inbar Levy, Andrew Christie, and Timothy Baldwin. 2021. Automatic resolution of domain name disputes. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 228–238.

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, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Michihiro Yasunaga, Xinyun Chen, Yujia Li, Panupong Pasupat, Jure Leskovec, Percy Liang, Ed H Chi, and Denny Zhou. 2023. Large language models as analogical reasoners. *arXiv preprint arXiv:2310.01714*.

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

Fangyi Yu, Lee Quartey, and Frank Schilder. 2023. Exploring the effectiveness of prompt engineering for legal reasoning tasks. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13582–13596.

Ruizhe Zhang, Qingyao Ai, Yueyue Wu, Yixiao Ma, and Yiqun Liu. 2023. Diverse legal case search. *arXiv preprint arXiv:2301.12504*.

A Error Analysis

Table 4 reports the basic statistics for the error types we observed in our error analysis.

Type	Frequency	Example
FP: Incorrect Reasoning	16.67% (1/12)	Carlos enjoys riding his skateboard in the skate park...
FP: Shared the same introduction and question pair.	25% (3/12)	My dog Louisa loves to learn new tricks...
FP: Lots of Similar Language	8.33% (2/12)	While making Eggplant Parmesean...
FP: Other	25% (3/12)	...
FN	25% (3/12)	...

Table 4: Manual categorization of error types. False positives are categorized as either "Incorrect Reasoning," "Shared the same introduction and question pair," "lots of similar language," or "Other." We use a single FN category because of the lack of errors to analyze.