If You Give an LLM a Legal Practice Guide

Colin Doyle*
colin.doyle@lls.edu
Loyola Law School
Los Angeles, California, USA

Aaron D. Tucker*†
aaron@far.ai
FAR.AI
Berkeley, California, USA

Abstract

Large language models struggle to answer legal questions that require applying detailed, jurisdiction-specific legal rules. Lawyers also find these types of question difficult to answer. For help, lawyers turn to legal practice guides: expert-written how-to manuals for practicing a type of law in a particular jurisdiction. Might large language models also benefit from consulting these practice guides? This article investigates whether providing LLMs with excerpts from these guides can improve their ability to answer legal questions. Our findings show that adding practice guide excerpts to LLMs' prompts tends to help LLMs answer legal questions. But even when a practice guide provides clear instructions on how to apply the law, LLMs often fail to correctly answer straightforward legal questions – questions that any lawyer would be expected to answer correctly if given the same information. Performance varies considerably and unpredictably across different language models and legal subject areas. Across our experiments' different legal domains, no single model consistently outperformed others. LLMs sometimes performed better when a legal question was broken down into separate subquestions for the model to answer over multiple prompts and responses. But sometimes breaking legal questions down resulted in much worse performance. These results suggest that retrieval augmented generation (RAG) will not be enough to overcome LLMs' shortcomings with applying detailed, jurisdiction-specific legal rules. Replicating our experiments on the recently released OpenAI o1 and o3-mini advanced reasoning models did not result in consistent performance improvements. These findings cast doubt on claims that LLMs will develop competency at legal reasoning tasks without dedicated effort directed toward this specific goal.

CCS Concepts

• **Applied computing** → **Law**; • **Computing methodologies** → **Natural language processing**; **Knowledge representation and reasoning**; • **Social and professional topics** → *Automation*.

*Equal contribution
†Work also conducted within the Department of Computer Science, Cornell University, Ithaca, New York, USA

CSLAW '25, München, Germany

ACM ISBN 979-8-4007-1421-4/25/03
https://doi.org/10.1145/3709025.3712220

Keywords

Law, Large Language Models, Propositional Logic, Retrieval Augmented Generation, Reasoning Models

ACM Reference Format:
Colin Doyle and Aaron D. Tucker. 2025. If You Give an LLM a Legal Practice Guide. In *Symposium on Computer Science and Law (CSLAW '25), March 25–27, 2025, München, Germany*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3709025.3712220

1 Introduction

Despite being trained on vast corpuses of data, LLMs often struggle to correctly answer questions that depend upon knowledge of domain-specific information. Retrieval-augmented generation (RAG) has emerged as a method for improving LLM performance at question answering by grounding LLM responses with accurate domain information [22]. With RAG, information is retrieved from a knowledge database and then injected into the prompt given to an LLM. RAG pipelines function like an open-book exam, giving the model a chance to answer a question using retrieved information.

RAG *ought* to help LLMs answer legal questions [1]. One noted shortcoming of LLM performance with legal reasoning tasks is a lack of knowledge about jurisdiction-specific rules and precedent [24]. With American law, all fifty states have their own, independent state constitutions, statutes, rules of procedure, and governing caselaw. Although the current generation of LLMs have been trained on enormous datasets, LLMs still struggle to properly answer legal questions that depend upon jurisdiction-specific rules [13]. RAG pipelines should be able to provide LLMs with exactly this type of granular, jurisdiction-specific legal information.

Whether LLMs can effectively use that information is another question [24]. RAG pipelines are most successful when retrieving factual information that an LLM can copy into a response. But in a legal context, RAG pipelines do not retrieve *information* for an LLM to *repeat* so much as they retrieve legal *rules* for an LLM to *apply*. Legal questions are often complex with multiple related parts and conditional logic. Even if an LLM is informed of the correct legal rules to apply to a set of facts, the scope or complexity of those rules may overwhelm an LLM's capacity to competently apply those rules [10].

This article examines how providing LLMs with relevant information from legal practice guides affects their performance at answering legal questions. Legal practice guides are a type of legal reference that helps attorneys become acclimated to a legal practice area without having to build up that understanding from scratch by reading troves of statutes, regulations, and legal opinions [14]. Practice guides are an ideal use case for evaluating RAG's potential for improving LLM performance at answering legal questions because practice guides are as straightforward and plainly written as legal documents get. Compared to other legal texts, practice guides

are clear and succinct, and they are already structured as step-by-step instructions for attorneys to follow. With other kinds of legal documents, the verbosity and complexity of the retrieved material might cause an LLM to provide erroneous responses. Experiments designed to test LLMs' capacity to apply legal principles based on retrieved information may instead become experiments testing LLMs' capacity to parse lengthy, confusing documents. Practice guides provide as clean as possible of an opportunity to observe how LLMs fare at applying retrieved legal rules to practical scenarios.

If existing practice guides can improve LLM performance at legal tasks, this has significant implications for LLMs' potential for performing legal work in the near future. Thousands of practice guides have already been written, covering virtually every area of law for every jurisdiction in the United States. Practice guides may be a bountiful resource for increasing LLM performance on technical, domain-specific legal tasks as an alternative or supplement to more expensive, time-consuming processes like finetuning models on jurisdiction-specific caselaw.

2 Background

2.1 Legal Background

We chose to test LLM performance answering legal questions in three different legal practice areas in three different U.S. jurisdictions: California law governing the tort law doctrine of res ipsa loquitur, Minnesota law governing the state's power of eminent domain, and New Jersey criminal law concerning pretrial incarceration. Each of these practices areas includes state-specific rules that make legal questions on these topics difficult for current LLMs to answer. These practice areas were selected to test LLM performance in a variety of jurisdictions and legal subject matter areas. Each practice area belongs to a different state in a different region of the country, and each concerns a different kind of litigation: private civil litigation that depends on common law rules; public law litigation rooted in statutory law, constitutional law, and state constitutional law; and procedural rules in criminal law that depend upon recently enacted state statutes and a state constitutional amendment.

California Res Ipsa Loquitur. California's tort law doctrine of res ipsa loquitur is a set of common law rules concerning the plaintiff's evidentiary burden in a negligence cause of action [34]. In an ordinary negligence lawsuit, a plaintiff must prove that the defendant breached a duty of reasonable care to the plaintiff. The plaintiff must establish what would have constituted reasonable care under the circumstances and how the defendant failed to exercise reasonable care. For cases in which the defendant's negligence can be readily inferred based on the plain facts of the case, the doctrine of res ipsa loquitur makes the plaintiff's job easier. Latin for "the thing speaks for itself," res ipsa loquitur creates a presumption of negligence that the defendant has the opportunity to rebut. Under California law, for res ipsa loquitur to apply: (1) the accident must be of a kind which ordinarily does not occur in the absence of someone's negligence; (2) it must have been caused by an agency or instrumentality within the exclusive control of the defendant; and (3) the accident must not have been due to any voluntary action or contribution on the part of the plaintiff [8]. Res ipsa loquitur is a state law doctrine, but it's a foundational legal concept taught to every first-year law student, and the doctrine exists in every state in the United States with only slight variation from jurisdiction to jurisdiction.

Minnesota Eminent Domain. Minnesota state law governing eminent domain is a combination of state statutory law, state constitutional law, and federal constitutional law [31]. Each of these sources of law restrict the government's power to seize private property. The U.S. Constitution sets the floor for individual rights protections against state use of eminent domain power across the United States. The government can seize private property only if the state takes that property for a public purpose and provides the original property owner with just compensation for the taking. LLMs are likely to be able to answer questions about the broader federal legal backdrop and eminent domain restrictions that apply across the United States. But many states offer property owners greater protection from government takings than the federal constitution provides. Minnesota is one of those states. Compared to federal takings law, Minnesota law has a narrower interpretation of what kinds of takings can constitute a "public purpose," and Minnesota has an additional "necessity" requirement that is absent at the federal level. Minnesota also has very specific statutory rules for when a property can be classified as blighted, abandoned, or environmentally contaminated — thereby justifying a government taking. These laws regarding eminent domain are unlikely to be well represented within LLM training data because the legal rules apply only to the state of Minnesota and differ from the rules of other states.

New Jersey Pretrial Detention. New Jersey pretrial incarceration laws are a set of a procedural rules that must be followed for the state to legally incarcerate a person after arrest but before trial in a criminal case [16]. These rules are specific, detailed statutory and state constitutional provisions that were enacted in New Jersey in 2017 as comprehensive criminal justice reform legislation. In New Jersey, criminal defendants cannot be detained pretrial on unaffordable money bond. Rather, a prosecutor must motion for a pretrial detention hearing. At the hearing, the defendant is entitled to be represented by an attorney who is allowed to cross-examine government witnesses and present evidence. To incarcerate someone pretrial, a court must consider a specific set of factors and make written findings concluding that no set of conditions of release would ensure the safety of the community, guarantee that the defendant would return to court, or prevent the defendant from obstructing the judicial process. These rules are specific to the state of New Jersey and are not applicable to other states, most of which do not have these detailed procedural rules governing pretrial incarceration.

2.2 LLMs and Legal Reasoning

Prompt Engineering. In recent years, large language models have shown a remarkable improvement in performance on natural language processing tasks [9]. A wave of contemporary research focuses on improving large language models' performance at complex tasks through novel prompting techniques. These techniques can encompass both prompt engineering and prompt architecture.

Prompt engineering refers to techniques for writing individual prompts to elicit LLM responses that exhibit stronger performance at natural language tasks. Prompt architecture refers to methods for structuring multiple prompts and responses to better accomplish complex tasks through a series of LLM calls. In contrast to conventional methods for improving LLM performance, prompting techniques do not require extensive retraining or fine-tuning, making these methods both cost-effective and widely accessible [36]. The prompting technique adopted in this paper, Chain-of-Thought (CoT) prompting, guides an LLM to develop its reasoning step-by-step before reaching an ultimate conclusion [37]. The goal of many of these techniques is to encourage an LLM to mimic the steps of multi-step thinking and reasoning that humans perform. But even state-of-the-art methods are limited. Their performance gains are limited to specific examples, and "concepts related to the LLM reasoning are not well-defined, hindering effective design of new more powerful schemes." [5].

Retrieval Augmented Generation. Retrieval augmented generation is designed to improve LLM performance at knowledge-intensive tasks by retrieving and injecting relevant information into the prompt given to the language model [22]. RAG has proven to be very effective when the retrieved information contains the correct answer to the question that the LLM has been asked to answer [35]. RAG pipelines have become an essential part of building out LLM software such as customer service chatbots that can answer questions that depend upon information specific to the business or the customer.

RAG for answering legal questions is more complicated than typical RAG use cases. A RAG system can retrieve relevant legal information, but the retrieved information rarely contains the answer to a legal question. Rather, the retrieved information contains legal rules or principles that must be applied to the facts of the case at hand to arrive at the correct answer to a legal question. If applying law to facts was simple, law school would be easier and lawyers' hourly rates would be lower. Accurately applying a legal rule may require a host of different legal skills, including careful reading of technical terms, navigating layers of conditional logic and exceptions (which may include applying layers of legal rules within legal rules), handling open-textured provisions like multi-factor tests, and incorporating practical knowledge about the world and the judicial system, to name a few.

Reasoning Models. Another approach to increasing an LLM's performance at complex reasoning tasks is by improving its ability to reason. LLM chatbots are trained to respond to user feedback in a preferred way [29] often called helpfulness [3]. This training method is based on asking people to rank the responses that a model generated. In contrast, Chain of Thought tries to improve performance by eliciting the model to generate intermediate "reasoning" steps [38]. Intuitively, a person might answer a question better by breaking it into multiple steps and working through them, rather than trying to answer them immediately. Mechanically, since an LLM uses a fixed amount of computation per each token,[1] answering after many tokens rather than immediately allows the LLM to use more computation in order to answer the question. This strategy has been found to be highly scalable [27] when models are trained to take advantage of this [20], though this can require exponentially more compute.

Propositional Logic. Propositional Logic is a classic topic in Philosophy and Computer Science, which analyzes the truth value of different combinations of logical statements. For this paper, we only need `NOT`, `OR`, and `AND`. The statement "`NOT` A" (denoted $\bar{A}$) is true if A is false, and false if A is true. The statement "A `AND` B" (denoted $A \wedge B$) is true if A and B are both true, and false otherwise. The statement "A `OR` B" (denoted $A \vee B$) is false if both A and B are false, and true otherwise.

A	B	$\bar{A}$	$A \wedge B$	$A \vee B$
False	False	True	False	False
False	True	True	False	True
True	False	False	False	True
True	True	False	True	True

Both $\wedge$ and $\vee$ are associative, meaning that

$$(A \wedge B) \wedge C = A \wedge (B \wedge C) = A \wedge B \wedge C,$$

$$(A \vee B) \vee C = A \vee (B \vee C) = A \vee B \vee C.$$

These operations can also be extended into probabilistic settings. If we take True to be the value 1 and False to be the value 0, then a proposition which is true with probability p is has the value p. `NOT` can then be implemented as $\bar{p} = (1 - p)$, while `AND` is $p \wedge q = p * q$. `OR` is more complicated, since if A is true with probability p and B is true with probability q, then if A and B are independent events then the probability of p or q is $p + q - p * q$ to avoid double-counting the possibility that they are both true. For example, the probability that at least one of two fair coin flips is heads is 75%. Using the fact that $A \vee B = \overline{\bar{A} \wedge \bar{B}}$, we have

$$p \vee q = \overline{\bar{p} * \bar{q}} = \overline{(1-p) * (1-q)} = 1 - (1-p) * (1-q).$$

In law, propositional logic is typically represented as necessary or sufficient conditions for a legal test or rule. If a condition is sufficient, the legal test is satisfied *if* the condition is true. If a condition is necessary, the legal test is satisfied *only if* the condition is true. In other words, if a condition is necessary, the legal test is not satisfied unless the condition is true, but the condition being true does not necessarily mean the test is satisfied. Legal tests, including the ones evaluated in this paper, often include multiple conditions. For example, under Minnesota state law, a government taking of private property is lawful only if three necessary conditions are true: the government had necessity for the taking; the taking was for a public purpose; and the government paid just compensation [16]. The public purpose requirement can be satisfied if any of three sufficient conditions is true: the land was taken for use by the general public or a public agency; the land was taken for use by a public service corporation; or the land was taken to mitigate harms of blight, environmental contamination, abandoned property, or public nuisance.

Although law is suffused with the language of formal logic, law is a "scruffy" domain, heavily experience-based and example-driven [32]. Legal reasoning is only pseudo-formalized, with a large body of formal rules that are often ambiguous, contradictory, and incomplete. While many legal tests are framed as propositional logic, the logical evaluative criteria are commonly infused with qualitative

[1] Tokens are roughly words, though some words break into multiple tokens.

judgment [2]. And not all legal tests conform to a logical framework. Many legal concepts are open-textured, lack hard boundaries, and are littered with exceptions, explicit and implicit. "Multi-factor" and "totality of the circumstances" legal tests require judges to consider different conditions, none of which are necessary or sufficient. Critics of these kinds of tests claim that the tests produce indeterminate results because there is no "rule" to apply beyond the judge's personal proclivities [33]. Proponents of these kinds of tests defend them on the grounds that they afford greater sensitivity to the facts of particular cases and avoid the arbitrary results of brightline rules. [19]. For this paper, LLM performance is not evaluated on "multi-factor" or "totality of the circumstances" legal tests because the legal conclusions under these tests tends to be more subjective and open to debate than tests with more rigidly specified necessary or sufficient conditions. LLMs have the potential to enable A.I. legal reasoning systems to process open-ended, qualitative legal inquiries that frustrated earlier rule-based systems, and LLM performance at applying more open-ended legal tests is worth investigating in future work [12].

Prior Work. For over four decades, the field of A.I. and law has developed computational models of legal reasoning for both theoretical and practical purposes. [4]. The LLM breakthrough has inspired a new wave of empirical scholarship that investigates both LLM performance at legal reasoning tasks and also the possibilities for integrating LLM technology with classic rule-based systems. [30].

Large language models challenge our expectations about what computers can do, how they should act, and what kind of mistakes they're prone to make. Across different domains, LLMs' capabilities form a "jagged frontier" — where they excel at some tasks — like writing boilerplate emails — and fail at others — like solving math problems. [15]. This jagged frontier of unpredictable performance is present with legal reasoning as well. Depending on the task it's assigned, an LLM might deliver a precise and persuasive legal argument, or it might hallucinate imaginary cases and stumble over basic logical deductions. Some research finds that LLMs can outperform law students on law school exams and perform only slightly worse than expert tax lawyers at answering tax questions. [11, 26]. But when presented with statutory language and asked to answer straightforward legal questions, GPT-3 was shown to regularly make mistakes with basic statutory reasoning. [7] Likewise, LLMs struggle to handle basic text tasks done in legal practice, although fine-tuning a model can greatly improve performance. [6]. At the time of writing, it remains an open question whether LLMs have developed an emergent legal reasoning ability, how much of LLMs' success at legal reasoning tasks is dependent upon similar examples within the LLMs' training data, and how much an LLMs' strong performance with one type of legal task is replicable with other legal tasks in other domains.

3 Methods

Datasets. For datasets, we chose three practice guides covering different areas of law from different U.S. jurisdictions: a California civil practice guide for tort law, a Minnesota practice guide for real estate law, and a New Jersey practice guide on criminal procedure. [16, 31, 34]. Within each guide, we selected a particular legal topic to test: for California, the tort law doctrine of res ipsa loquitur; for Minnesota, state statutory and constitutional law governing eminent domain; and for New Jersey, state statutes concerning pretrial detention procedures.

For each topic, we extracted from the practice guides any relevant instructions for answering legal questions on that topic. These excerpts simulate a best case scenario for information retrieval, allowing us to measure the model's performance given that the correct part of a practice guide has been included within the prompt.

For our experiments on *real cases*, we manually extracted the facts and holding of each relevant case referenced within that part of the practice guide. For each case, the facts provide background information on the parties and the legal dispute, and the holding provides the legal conclusion for that case along with the reasoning behind that conclusion. This resulted in 12 California res ipsa cases and 5 Minnesota takings cases. Since the New Jersey pretrial detention reforms went into effect in 2017 and have not yet produced a substantial body of caselaw, we did not include real cases.

For our experiments on *hypothetical cases*, we wrote hypothetical examples to address different aspects of the legal rules contained in the practice guide. We had 13 hypotheticals for California Res Ipsa Loquitur, 20 hypotheticals for Minnesota Eminent Domain, and 14 hypotheticals for New Jersey pretrial detention. We annotated each example with the correct overall legal conclusion, along with the correct conclusion for each legal subissue. In general, law is not deterministic. Strong legal arguments are usually better understood as being "persuasive" rather than "correct." [18]. These hypotheticals were purposefully written to be "easy" legal questions that had clearly correct and incorrect answers. They would make terrible law school exam questions. The hypotheticals were also purposefully written as simplified versions of the facts of a legal case so that the experiments would measure an LLM's ability to follow instructions for applying law to facts, not an LLM's ability to parse complicated fact patterns. Future work should test LLM performance with more complicated facts and trickier legal questions. For this paper, we wanted to capture a baseline level of performance.

Two hypotheticals on Minnesota law are representative of the type of hypothetical fact patterns that we designed. The first example is a straightforward application of the government's use of imminent domain that does not run afoul of any federal or state legal rules:

> The city of St. Paul, Minnesota recently exercised its power of eminent domain to take possession of six residential properties in the Thomas-Dale neighborhood of the city. St. Paul acquired the property to raze the homes and expand the width of a highway running alongside the neighborhood. The city of St. Paul compensated the property owners for the fair market value of their property.

In this second example, the government's use of imminent domain would run afoul of Minnesota-specific imminent domain rules. Minnesota has a "necessity" requirement, which means that the state cannot take property for "[s]peculative purposes" such as unspecified future development.

> The city of St. Paul, Minnesota recently exercised its power of eminent domain to take possession of all

of the homes within a residential city block. Within this city block, 70 percent of the buildings had significant building code violations and the cost to repair or restore the buildings would exceed more than 70 percent of those buildings' estimated market value. The city of St. Paul compensated the homeowners for the fair market value of their properties. The city has decided to leave the property as-is for the time being, and the residents will be allowed to stay in their homes. At an as yet unspecified time in the future, the city may decide to sell the property or redevelop it for another use.

Prompting. We used three different prompt templates to evaluate LLM performance at answering legal questions. The first prompt template (*Fact*) served a control to establish the LLM's baseline performance absent any help from the practice guide. (*Fact*) provided the LLM with only the facts of the case, without information from the practice guide. The second prompt template (*+Guide*) provided the LLM with the facts of the case and the excerpt from the practice guide. For these prompt templates we took advantage of probabilistic LLM outputs by requesting 10 responses from the LLM for each query for each proposition or question, and then averaging over the results.

The final prompt template (*Prop.*) broke the excerpt of the practice guide down into distinct components based on different parts of the relevant legal rule. A separate LLM query was made for each part of the legal rule, and the LLM was asked to evaluate whether that part of the legal rule was met. We then combined the different propositions into an overall score for the hypothetical using the following definitions in propositional logic.

Res Ipsa:

the accident was...
of a kind which ordinarily does not occur in the absence of someone's negligence
∧ caused by an agency or instrumentality within the exclusive control of the defendant;
∧ not due to any voluntary action or contribution on the part of the plaintiff

Minnesota Eminent Domain:

The taking was necessary
∧ the government paid just compensation for the taking
∧ the taking was for a public purpose: the taking was for... (
the possession, occupation, ownership, and enjoyment of the land by the general public or by public agencies
∨ the creation or functioning of a public service corporation
∨ the mitigation of a blighted area
∨ the remediation of an environmentally contaminated area
∨ the reduction of abandoned property
∨ the removal of a public nuisance
)

New Jersey Pretrial Detention:

There was a lawful, valid justification for the judge to order the defendant to be detained pretrial
∧ The court followed the correct procedures (
The defendant who was detained pretrial was eligible for pretrial detention
∧ The defendant was granted a pretrial detention hearing within the timeframe required by law
∧ The pretrial detention of the defendant was the result of legally required motion and hearing
∧ The court considered the correct factors when it decided to detain the defendant pretrial
∧ The pretrial detention hearing adhered to due process requirements
)

We had two variants of the propositional strategy. *Prop. 2* broke down the prompt into only the first-level propositions and combined all of the LLM subqueries for the proposition into a single LLM query. *Prop. 3* used the full propositional structure outlined for the prompt.

Close Reading of Predictions. We conduct a hybrid quantitative and qualitative study of LLM performance on legal questions.

Quantitatively, we follow methods similar to machine learning by assembling a dataset, collecting labels, then calculating the accuracy of several different methods of predicting the labels for the datapoints. We empirically study the Computer Science-oriented question of how well RAG improves performance on an LLM task, along with the LLM-related questions of which prompting strategies work best.

Qualitatively, we incorporate techniques from legal scholarship, including close reading and analysis of legal arguments, to examine the responses that the LLMs generated. Rather than focusing only on whether the LLMs arrived at the right answer, we also analyze the legal reasoning that supports those conclusions. When an LLM fails to arrive at the correct legal answer, a close reading of its responses can reveal the particular errors in legal reasoning that led the model astray.

4 Results on Real Cases

4.1 California Res Ipsa Loquitur

	Facts	+Guide	Prop. 2
GPT-3.5	0.67	0.70	0.57
GPT-4	0.72	0.71	0.78
GPT-4o	0.84	**0.91**	0.81
Claude Haiku	0.61	0.69	0.63
Claude Sonnet	0.75	0.66	0.50
Claude Opus	0.76	0.74	0.67
Sonnet 3.5	0.74	0.74	0.72

Figure 1: Accuracy on Real California Res Ipsa Loquitur Cases

When asked whether the doctrine of res ipsa loquitur applied to the facts of real cases, the LLMs had some ability to discern the correct answers based on the facts of the cases alone, and their performance often improved when given a relevant excerpt from a practice guide. This improvement is slight across most models, the most being an 8% improvement for Claude Haiku. Other models

did not have stronger performance when given the practice guide. Claude Sonnet's performance with the practice guide was 9% worse than with facts alone.

Breaking down the legal test into component parts resulted in worse performance across almost all models, which was unexpected. GPT 3.5 is good example. The model acheived 67% accuracy with just the facts and improved to 70% accuracy with the practice guide but dropped to 57% accuracy when the practice guide was broken up into the different components of the legal test for res ipsa loquitur. The transcripts of the LLM responses reveal why performance tended to drop. Breaking up res ipsa loquitur into the component parts of the legal test and asking the LLM to answer each component in a separate API call sometimes resulted in the LLM overthinking the question. The LLMs had a tendency to get lost in the details of the facts of the case or the intricacies and possible exceptions to the rule, which led to incorrect conclusions. At the guide level, the LLMs more consistently arrived at the straightforward, correct legal answer.

The case of *Baumgardner v. Yusuf* illustrates the trend. In this case, a sponge was left inside a patient's leg during surgery and the patient sued the surgeon for negligence under a res ipsa loquitur theory. A legal question in the case is whether the surgeon had exclusive control over the instrumentality of harm given that the nurses who were assisting him were in charge of the sponges. At the *+Guide* level, ChatGPT-3.5 correctly notes "The surgical procedure, including the use and counting of sponges, was within the control of Dr. Yusuf and the assisting staff. While Dr. Yusuf argued that the sponge count procedure was under the control of the hospital nursing staff, the ultimate responsibility for the operation and ensuring no foreign objects are left in the patient lies with the surgeon." But at the *Prop. 2* level, ChatGPT-3.5 is led astray by a prolonged consideration of Dr. Yusuf's testimony at trial, which convinces the LLM that the surgeon did not have exclusive control.

4.2 Minnesota Eminent Domain

	Facts	+Guide	Prop. 2	Prop. 3
GPT-3.5	0.98	0.80	0.87	0.91
GPT-4	0.96	0.86	0.44	0.44
GPT-4o	0.98	0.84	0.68	0.72
Claude Haiku	**1.00**	0.82	0.69	0.73
Claude Sonnet	0.94	0.72	0.36	0.43
Claude Opus	0.88	0.64	0.35	0.44
Sonnet 3.5	0.94	0.84	0.40	0.56

Figure 2: Accuracy on Real Minnesota Eminent Domain Cases

When asked whether a government taking was lawful given the facts of real cases, the LLMs were consistently able to discern the correct answers based on the facts of the cases alone. Counter-intuitively, providing the LLMs with relevant excerpts from practice guides harmed performance, even though eminent domain law in Minnesota has substantial state-specific variation. In these experiments, we additionally broke down the *Prop.* method into two levels. *Prop. 2* breaks the legal test for eminent domain down into multiple questions, and *Prop. 3* breaks some of those questions down into subquestions. Breaking down the legal tests tends to compound the negative effect of the practice guide and results in the worst performance overall. This negative effect is most pronounced on the advanced models (Sonnet, Sonnet 3.5 and GPT-4) whereas the less advanced models (GPT-3.5, Haiku) see either minor decreases or increases in performance.

Qualitative analysis of the transcripts of the LLM responses can show how the LLMs' legal reasoning went off track with these methods. When the legal test is decomposed and the LLM is asked to answer a subquestion of the test, the LLMs have a tendency to miss the forest for the trees. The LLMs focus on obscure parts of the practice guide that are not relevant to the fact pattern of the hypothetical. On many occasions, the LLMs mistakenly interprets the absence of factual information concerning a theoretically possible unlawful government action as proof that the government took that unlawful action.

4.3 Limitations of using real cases

In the course of our work, we found that predicting the outcomes of real cases was less straightforward of an experiment than we had initially anticipated.

First, since legal opinions are persuasive texts, legal opinions often do not have a clean separation between the facts of the case and the legal conclusions drawn from applying legal rules to those facts. This raises two issues. One, the facts are often intermixed with legal reasoning about those facts. Two, the facts are often characterized in a way that supports the legal conclusions that the court draws from those facts. Although we tried to extract "clean" versions of the facts of each case without rewriting parts of the original text, the source material makes it more difficult to discern how much an LLM's legal conclusions are the product of the LLM's legal reasoning skills as opposed to the LLM's ability to extract legal conclusions from contextual clues about how facts have been characterized.

Second, given that the facts come from appellate cases, these cases tend to concern difficult, thorny legal questions about gray areas of the law. In contrast, practice guides tend to be concerned with the everyday routine application of the law — not the less common cases in which the legal outcome is uncertain and reasonable minds may differ. This makes the legal outcomes of these cases less clearly useful as a "ground truth" label of whether or not a legal practice guide is helpful for an LLM applying law to facts.

Third, the legal opinions were from appellate courts that sometimes defer to prior lower court findings or would remand the case to a lower court to reach a final conclusion on a legal issue. For example, many of the res ipsa cases cited by our practice guide were appellate cases in which the appellant claimed that a jury should have been instructed on res ipsa. Even when the appellant won the appeal, the result was rarely that the appellate court ruled that res ipsa applied. Rather, the appellate courts would rule that res ipsa could have applied and therefore the case would be remanded to the trial court for a new trial for a jury to make the final determination of whether res ipsa did apply. This makes it harder to evaluate LLM legal reasoning using real case outcomes because those case

outcomes did not offer clear yes-or-no legal conclusions based on a set of facts.

5 Results on Hypothetical Cases

We also evaluated the model performance on a set of newly-written hypothetical cases.

5.1 California Res Ipsa Loquitur

	Facts	+Guide	Prop. 2
GPT-3.5	0.52	0.67	0.71
GPT-4	0.55	0.63	0.82
GPT-4o	0.85	0.84	0.85
Claude Haiku	0.46	0.60	0.76
Claude Sonnet	0.59	0.63	0.72
Claude Opus	0.86	0.87	0.86
Sonnet 3.5	0.76	0.88	**0.91**

Figure 3: Accuracy on California Res Ipsa Loquitur Hypotheticals

For California res ipsa loquitur hypotheticals, the general trend was that the LLMs performed better when given the practice guide than when given only the facts of the case and performed best when the legal test for res ipsa loquitur was decomposed into three subtests. But some models' performances hardly varied across the different setups (GPT-4o and Claude Opus). Analysis of the transcripts of the LLMs' responses reveals that, when given only the facts of the hypothetical, the models tended to overlook the nuances of the three parts of the res ipsa loquitur test and leap to conclusions. When given the practice guide and when the legal test was broken down into component parts, the LLMs spent more time analyzing the issues, which often led to the correct conclusions.

In some instances, the LLM was able to connect the facts of the case with conditions mentioned in the practice guide to arrive at the correct legal conclusion. For example, Sonnet-3.5 was able to use information from the practice guide to correctly assess a hypothetical in which a plaintiff fell at a restaurant and sued for negligence: "In this case, we're dealing with a slip-and-fall accident. The practice guide specifically mentions that 'res ipsa loquitur is not well suited for slip-and-fall cases because no inference of negligence can arise simply upon proof of a fall on the defendant's floor.' This suggests that this element may not be met."

5.2 Minnesota Eminent Domain

	Facts	+Guide	Prop. 2	Prop. 3
GPT-3.5	0.55	0.80	0.78	0.79
GPT-4	0.65	0.88	0.85	0.77
GPT-4o	0.61	0.93	0.91	0.78
Claude Haiku	0.73	0.81	0.73	0.76
Claude Sonnet	0.74	0.91	0.76	0.78
Claude Opus	0.74	0.93	0.95	0.87
Sonnet 3.5	0.88	0.95	**0.96**	0.85

Figure 4: Accuracy on Minnesota Eminent Domain Hypotheticals

	Facts	+Guide	Prop. 2	Prop. 3
GPT-3.5	0.32	0.77	**1.00**	0.98
GPT-4	0.26	0.80	0.97	0.76
GPT-4o	0.17	0.91	0.99	0.70
Claude Haiku	0.56	0.79	**1.00**	**1.00**
Claude Sonnet	0.62	0.97	0.98	0.99
Claude Opus	0.50	0.86	0.99	0.82
Sonnet 3.5	0.83	0.90	0.98	0.74

Figure 5: Accuracy on Minnesota-Specific Eminent Domain Hypotheticals

For Minnesota eminent domain hypotheticals, the LLMs tended to perform best when given the practice guide. Claude Opus and Sonnet 3.5 achieved the best results overall with *Prop. 2*, the first layer of decomposition.

	Facts	+Guide	Prop. 2	Prop. 3
GPT-3.5	0.74	0.82	0.59	0.63
GPT-4	0.98	0.95	0.75	0.78
GPT-4o	0.98	0.95	0.85	0.85
Claude Haiku	0.88	0.84	0.52	0.55
Claude Sonnet	0.84	0.86	0.58	0.60
Claude Opus	0.93	0.98	0.91	0.91
Sonnet 3.5	0.91	**0.99**	0.95	0.95

Figure 6: Accuracy on Non-Minnesota-Specific Eminent Domain Hypotheticals

Figures 5 and 6 offer another view on the data captured in Figure 4 on Minnesota eminent-domain hypotheticals. Figure 5 concerns a subset of hypotheticals in which knowledge of Minnesota-specific law is necessary for arriving at the correct legal conclusion. Figure 6 concerns a subset of hypotheticals in which general knowledge of eminent domain law across the country would be sufficient for arriving at the correct legal conclusion. Surprisingly, while *Prop* methods obtain the best performance on the Minnesota-specific questions, *+Guide* still gets the best performance on non-Minnesota-specific questions for most Claude models despite the fact that there is no need for Minnesota-specific information. For the GPT models and Claude Haiku, *Facts* gets the best performance.

For the Minnesota-specific hypotheticals based on facts alone, the OpenAI models, GPT-3.5 and GPT-4, perform very poorly, far below random chance at guessing a "yes" or "no" question correctly. At the same task, the Anthropic models all scored similar to random chance. Analysis of the transcripts of the LLMs's responses reveals that the OpenAI models' accurate knowledge of federal law betrayed the models when answering questions for state law that had different requirements. The models either applied the federal standard or hallucinated a state standard that matched the federal standard to consistently arrive at the incorrect answer. Meanwhile, the Anthropic models' less accurate knowledge of federal law meant that their performance was closer to random chance. These models would invoke the federal legal precedent but not always apply it correctly. So even when they arrived at the right answer half of the time, it was seldom for the correct reason.

Another trend is that with non-Minnesota specific hypotheticals, GPT-3.5, Haiku, and Sonnet all perform significantly better when given the practice guide (*+Guide*) than when the legal test is decomposed (*Prop. 2* and *Prop. 3*). The transcripts of the LLM responses reveal that the LLMs made the same errors with the Minnesota hypotheticals as they did with the real Minnesota cases. When the LLM looks at the problem more closely under *Prop. 2* or *Prop. 3*, it latches onto more obscure rules and interprets the absence of factual information about a theoretically possible legal violation as evidence of that legal violation. By breaking the legal question up into its constituent parts, the LLM zooms in too far, finding legal issues where they don't exist.

5.3 New Jersey Pretrial Detention

	Facts	+Guide	Prop. 2	Prop. 3
GPT-3.5	0.57	0.64	0.57	0.80
GPT-4	0.59	0.73	0.93	0.78
GPT-4o	0.59	0.68	0.86	0.73
Claude Haiku	0.56	0.61	0.64	0.62
Claude Sonnet	0.62	0.81	0.79	0.80
Claude Opus	0.78	0.82	0.93	0.79
Sonnet 3.5	0.78	0.88	**1.00**	0.91

Figure 7: Accuracy on New Jersey Pretrial Detention Hypotheticals

LLM performance with New Jersey pretrial detention hypothetical exhibits the opposite trend of what was found with the Minnesota hypotheticals. The LLMs' strongest performance is with the first level of decomposition *Prop. 2*. With Minnesota takings, when the LLM examines each subrule separately under *Prop. 2*, the LLM tends to make mistakes because it latches onto obscure issues in the practice guide and conjures up legal issues where they do not exist in the fact pattern. With New Jersey pretrial detention, the opposite is true. When the LLM is given the whole excerpt of the guide, the LLM tends to make mistakes because it *doesn't* latch onto issues that should determine the legal outcome, instead considering those issues as one factor among many. But when the legal test is broken up into parts for the LLM to examine separately, the LLM tends to perform better because it gives these issues the weight they deserve.

Chat-GPT-4's responses to a particular hypothetical can illustrate this trend. In this hypothetical, a judge purposefully sets a defendant's money bail at unaffordable amount to effectively keep the defendant in jail without explicitly ordering the defendant to be incarcerated. With the practice guide alone, GPT-4 misses this as an issue and finds the order of pretrial detention lawful because other requirements are satisfied. With *Prop. 2*, GPT-4 notices the issue, finding that "The court did not adhere to the legally required procedures because detaining Riley through the imposition of an unpayable money bail contradicts the intent and structure of the Criminal Justice Reform Act."

Decomposition is not always the answer. Although the LLMs tend to perform better with *Prop 3* than with *+Guide*, *Prop. 3* yields worse results than *Prop. 2* for all models except GPT-3.5. At this level of decomposition, the LLMs can focus too narrowly on one component of a legal test and miss broader cross-cutting issues that can affect the outcome for multiple parts of a legal test.

6 Reasoning Models: o1 and o3-mini

OpenAI's o1 and o3-mini reasoning models were released after this paper was accepted. We replicated our evaluations with these models and got more unexpected results. These models were trained via reinforcement learning to do reasoning [28] using a long sequence of text known a as chain of thought [38]. As such, one might expect them to be better at legal reasoning, however the gains over even only GPT-4o were often marginal. Because the models are trained to decompose questions and to save costs, we did not evaluate our propositional logic (*Prop. 2* and *Prop. 3*) methods. Similarly, the OpenAI API only allows users to request 8 generation at a time, so these evaluations use 8 samples for the *Facts* and *+Guide* strategies. For comparison, we report the best-performing GPT-4o score for the *Facts* and *+Guide* strategies.

	o1		o3-mini		4o
	Facts	+Guide	Facts	+Guide	
CA Res Ipsa	0.68	0.79	0.67	0.81	**.91**
MN Eminent Domain	0.83	0.53	**0.98**	0.78	**.98**

Figure 8: o1 and o3-mini Accuracy on Real Cases

Surprisingly, the o1 and o3-mini models do not improve over the best GPT-4o scores between the *Facts* and *+Guide* strategies. This suggests that while o1 and o3-mini were trained to do reasoning, they may have difficulty applying that reasoning in messier real-world domains. Further, the o1 and o3-mini models had sharply decreased performance in the *Facts* setting as compared to 4o. Similar to our previous results, neither the *Facts* nor *+Guide* strategy dominated the other.

	o1		o3-mini		4o
	Facts	+Guide	Facts	+Guide	
CA Res Ipsa	0.77	0.84	0.72	0.80	**.85**
MN Eminent Domain	0.72	**0.99**	0.60	0.95	.93
NJ Pretrial Detention	0.62	0.67	0.51	**0.69**	.68

Figure 9: o1 and o3-mini Accuracy on Hypothetical Cases

In keeping with the difficulty of applying LLM-based reasoning to real-world situations, the *+Guide* strategy always improved over the *Facts* strategy on hypothetical cases. However, o1 and o3-mini did not always improve over GPT-4o, getting a worse score on California Res Ipsa, a marginally better score on NJ Pretrial Detention, and only doing substantially better on MN Eminent Domain. This suggests that the improvements to the reasoning capabilities from this model series do not automatically extend to improved *legal reasoning* even on hypothetical questions.

7 Discussion

7.1 High Variability of Results

These experiments produced surprisingly variable results. Across different subject areas, each of the four methods for answering legal questions (*Facts*, *+Guide*, *Prop. 2*, and *Prop. 3*) achieved the highest accuracy for a particular model. None of the methods consistently produced stronger results across all of the models or legal subject areas. Using RAG to inject relevant information from legal practice guides tended to improve performance across all models but occasionally made no difference. From model to model and legal subject area to legal subject area, the *Prop. 2* and *Prop. 3* methods sometimes improved and sometimes hurt LLM accuracy compared to the practice guide alone. Consider Figure 4, Accuracy on Minnesota Eminent Domain Hypotheticals. For almost every model, the *Prop. 2* method was less accurate than giving the model the guide alone. But with Sonnet 3.5, the *Prop. 2* method not only produced the most accurate results for the model, it produced the most accurate results overall across all models and methods for that subject area.

Not only were results inconsistent, but they were often counterintuitive. For example, adding the practice guide helped the real California cases but not the real Minnesota cases. This is surprising, both because we expect the practice guide to lead to improvement, and because we expect the practice guide to be most helpful in domains with state-specific variation.

Surprisingly, increasing model capability does not reliably produce stronger results. For some legal doctrines like res ipsa loquitur, this may relate to the potential flexibility of the legal rule. As the model capability increases, LLMs may become increasingly creative in explaining why an accident might occur without negligence or what "exclusive control" means. Since res ipsa loquitur only applies if the "incident was of a type that does not generally happen without negligence" and the defendant was in "exclusive control" of the instrumentality of harm [21], more capable models may have been misled by their creative capacity to conjure up scenarios in which the incident could have happened in the absence of negligence and to ponder the definition of "exclusive control." That said, this experiment had a limited number of cases, and this observation may not extend to other fact patterns and legal questions.

The results are fairly different between real and hypothetical cases, even holding the legal domain fixed. In our hypothetical cases, adding the practice guide typically helped. In our real cases *+Guide* was always worse than *Facts* on Minnesota eminent domain but not on California res ipsa cases.

Why is the practice guide more useful on the hypotheticals than on the real cases? There are different explanations that future work could explore.

One theory is simply length: naively applying the practice guide does relatively poorly on the real Minnesota Eminent Domain cases, which have long fact sections compared to hypotheticals as well as a relatively long practice guide compared to Res Ipsa. This theory could be tested by seeing how performance varies while adding varying amounts of irrelevant text [25].

A second theory is that LLMs are trained on text found across the internet, and so they are better at applying the practice guide in situations that look like academic tests. In our experiments the hypotheticals often have worse overall performance within a domain, but the distribution of the hypotheticals and real cases are different. A useful experiment might be to compare performance on hypothetical versions of real cases to performance on the real cases.

A third theory is that the cases themselves are from quite different distributions. The hypotheticals were written to test how well the LLMs can handle each specific requirement of the practice guide, and so it makes sense that the practice guide would be more helpful.

7.2 Overall Error Rate

Separate from the relative performance between different models and prompts, the models performed poorly on the hypothetical cases in absolute terms. The hypothetical cases were written to be legally unambiguous questions that are clearly answered by the practice guide, so 100% accuracy should be achievable. One might expect that an LLM would make consistent legal reasoning errors based on a single legal misconception that is related to a subset of cases. Here, the LLM would produce incorrect results for the cases that present this legal issue. One might also expect an LLM to handle routine legal tasks easily, but sometimes makes mistakes regarding subtle or ambiguous points. Here, the LLM would reliably produce correct results for clearcut cases but struggle with cases that are in a legal gray area and are similarly challenging for humans. What we observed was different. The LLMs' errors were not related to consistent legal issues and were not related to more subtle or ambiguous points, but rather were seemingly random. For example, depending on whether a question was decomposed into two or three parts, the LLMs would sometimes make the surprising, egregious mistake of overlooking or dismissing critical details such as "the defendant was denied a lawyer" when answering whether a criminal proceeding that resulted in a defendant's incarceration had any procedural violations.

7.3 RAG and Decomposition

This project started as a basic sanity check of operationalizing a definition of a "good" practice guide as being one that improves case prediction performance when given to an LLM. In this original

idea, the expert-written practice guide was supposed to be a gold standard practice guide that we could compare other practice guides to. Instead, we discovered that providing the practice guide could *decrease* performance.

We were surprised to discover that decomposing legal questions into separate subquestions sometimes helped and sometimes hindered LLM performance. We had expected that decomposition would lead to better LLM performance because lawyers are taught to break down legal problems and address them methodically part-by-part and because LLM performance can falter when the LLM's context window has too much information for the LLM to reliably process. We were further surprised to discover that the effect of decomposition was inconsistent across legal subject areas. Sometimes even within one legal subject area, different decomposition approaches would achieve different results with different models. For example, with the New Jersey hypotheticals, each of the decomposition approaches (+Guide, Prop. 2, and Prop. 3) was responsible for the strongest performance for a different model.

Further it appears that, despite being trained to break up problems and reason through them, the *+Guide* strategy for the o1 and o3-mini models was outperformed by the *Prop. 2* or *Prop. 3* strategy using an older model for every task but Minnesota Eminent Domain hypotheticals. This shows that there is still an elicitation gap between the best performance attainable by hand-written strategies and implementations and straightforward usage of the o1 and o3-mini models, *even when* the hand-written strategy is as prone to systematization as "encode the practice guide into propositional logic, then evaluate the propositions separately".

A close reading of the LLM responses reveals some trends in the kinds of mistakes that LLMs would make based on how a question was broken down. When asking a legal question in whole resulted in a right answer and asking a legal question piece-by-piece resulted in a wrong answer, it was usually because the LLM found issues where there weren't any. Vice versa, when asking a legal question in whole resulted in a wrong answer and asking a legal question piece-by-piece resulted in a right answer, it was usually because the LLM overlooked an issue at the higher level and identified it at the broken-down level.

These errors may be the result of chat-based LLMs overvaluing the text given to them by a RAG pipeline. In the Prop. settings of the Non-Minnesota-Specific Eminent Domain Hypotheticals, providing broken-down practice guide information could make the LLM latch onto more obscure rules. This might be a consequence of training the LLM to function as a chatbot. LLM chatbots (such as the GPT series or Claude) are typically trained to be helpful assistants to a human user in a dialog setting [3]. This may decrease the LLM's suitability for highly technical and precise RAG settings, since when a person types a message to the LLM they are often trying to instruct the LLM to focus on certain information. With RAG, an LLM trained to be a helpful assistant might infer that retrieved information is highly relevant simply because it is part of the prompt. In these experiments, the LLMs may have identified non-existent legal issues in the facts based upon how important the legal issue seemed to be within the prompt rather than how much the facts of the case actually gave rise to this legal issue.

These are potentially large obstacles for automating legal work. Using a legal practice guide for RAG produces inconsistent results, and there is no consensus prompt decomposition strategy that achieves the best overall performance among the strategies we tried. Would-be-legal-automators may need to develop specific prompting strategies not just for every LLM, but for every legal domain and LLM pair that they use.

7.4 Robustness of Findings to New Model Releases

At the beginning of this project, only GPT-3.5, GPT-4, Claude Haiku, Claude Sonnet, and Claude Opus were available. GPT-4o and Claude Sonnet 3.5 were released, and then OpenAI o1 and o3-mini were released shortly before the final version of this paper. We were pleased (and surprised) to find that many points from our discussion remained relevant to the newer model releases. While GPT-4o and Claude Sonnet 3.5 tended to be straightforward improvements over their predecessors, the best prompting strategy still tended to vary between different tasks and models, and the newer chatbots were not always better. Surprisingly, o1 and o3-mini often did worse than GPT-4o, only leading to substantial improvements for Minnesota Eminent Domain hypotheticals.

8 Future Work

8.1 Unlocking legal reasoning with LLMs

Law is a unique field with particular concerns that won't be answered by Generative A.I. research in other domains. Most legal tasks don't neatly fit into the categories of the kinds of tasks where LLMs thrive or struggle. Law has one foot in language and one foot in logic. This makes the problem of LLM reliability more pronounced in law than other fields, because automation is only possible through language models and the cost of errors is high. Generative A.I. seems to have the potential to automate many legal tasks, but its capabilities have not been fully vetted.

A major concern this paper raises for the prospects of legal automation is that LLMs show a troubling sensitivity to question framing. Their answers to legal questions change dramatically based on whether and how a legal question is broken down. If breaking down a legal questions consistently helped or consistently hurt LLM performance, then our results would point to an approach for working with LLMs that would enable them to better approximate accurate legal reasoning. But the results are wildly inconsistent. LLMs can produce radically different answers to the same legal question based on minor variations in how the question is presented. This instability emerged even in our controlled experiments using short hypotheticals paired with practice guides containing explicit answers. The real world of legal practice is far messier than these contrived scenarios. Related work measuring LLM performance with math problems has likewise found that LLM performance varies dramatically based upon seemingly insignificant changes in how a question is phrased [25].

It's an open question whether different techniques could enable LLMs to perform legal reasoning capabilities that currently seem beyond their capabilities. The LLM shortcomings observed in this paper should be recognized as tendencies, not upper limits on what LLMs can accomplish. Whatever responses an LLM produces is necessarily a combination of an LLM's underlying capabilities, the prompts that a person has designed, and the architecture of

chaining prompts and responses together. When a series of LLM calls seems unable to accomplish a legal reasoning task effectively, it's difficult to pinpoint whether that limit is because of underlying deficiencies with LLMs, or if that reasoning task could be done more effectively with different prompting or different system architecture that segments or structures the task differently.

For example, the experiments in this paper used a fixed system prompt that included the text, "You are an expert legal assistant." How would the results vary with a different persona, or in a different setting? [17] Would the LLM answer questions differently if it were told that its client was the defendant or plaintiff? To decompose legal rules, we broke them down in the way that a lawyer would. Could LLMs be trained to better follow compositional instructions? Might there be more effective non-anthropomorphic ways of decomposing legal reasoning tasks? LLM performance at legal reasoning might also be improved using techniques such as process supervision, which has shown promising results in mathematical tasks [23].

8.2 CS&Law Scholarship

Both computer science and legal methodologies were critical to this project and are critical to understanding this domain. Without a legal analysis of the LLM transcripts, we would have an incomplete picture of how the LLMs succeeded or failed at legal reasoning tasks. Without computer science knowledge of LLM training details, we would lack technical theories for why the LLMs succeeded or failed at these legal reasoning tasks. Combining legal and computer science methodologies creates the opportunity for new insights that could not be achieved in either discipline in isolation.

9 Conclusion

Large language models struggle to answer highly specific legal questions. Could LLMs' performance improve if they are given access to relevant legal practice guides? Sometimes. Our findings suggest that injecting relevant excerpts from practice guides into prompts for LLMs tends to improve LLM performance at answering legal questions. But even when a practice guide provides clear instructions on how to apply the law, LLMs often fail to correctly answer straightforward legal questions. If a practice guide is used to structure a series of LLM queries that each analyze discrete issues which are then combined to answer a broader legal question, performance sometimes improves and sometimes becomes worse. Results vary considerably across models and legal subject areas. LLMs cannot be used as a drop-in replacement for lawyers even for straightforward tasks such as applying a legal practice guide to a clear set of facts. These findings have implications for the potential for generative A.I. to automate legal tasks, particularly through agentic systems and retrieval augmented generation [11].

Acknowledgments

Aaron Tucker was supported by scholarship funding from Open Philanthropy during early stages of this work, and the o1 and o3-mini experiments were supported by FAR.AI general support funds.

References

[1] Ayyoub Ajmi. 2024. Revolutionizing Access to Justice: The Role of AI-Powered Chatbots and Retrieval-Augmented Generation in Legal Self-Help. (2024).
[2] Dean Alderucci. 2020. The Automation of Legal Reasoning: Customized AI Techniques for the Patent Field. 58 (2020).
[3] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. 2022. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. arXiv:2204.05862 [cs.CL] https://arxiv.org/abs/2204.05862
[4] Trevor Bench-Capon, Michał Araszkiewicz, Kevin Ashley, Katie Atkinson, Floris Bex, Filipe Borges, Daniele Bourcier, Paul Bourgine, Jack G. Conrad, Enrico Francesconi, Thomas F. Gordon, Guido Governatori, Jochen L. Leidner, David D. Lewis, Ronald P. Loui, L. Thorne McCarty, Henry Prakken, Frank Schilder, Erich Schweighofer, Paul Thompson, Alex Tyrrell, Bart Verheij, Douglas N. Walton, and Adam Z. Wyner. 2012. A History of AI and Law in 50 Papers: 25 Years of the International Conference on AI and Law. 20, 3 (2012), 215–319. doi:10.1007/s10506-012-9131-x
[5] Maciej Besta, Florim Memedi, Zhenyu Zhang, Robert Gerstenberger, Guangyuan Piao, Nils Blach, Piotr Nyczyk, Marcin Copik, Grzegorz Kwaśniewski, Jürgen Müller, Lukas Gianinazzi, Ales Kubicek, Hubert Niewiadomski, Aidan O'Mahony, Onur Mutlu, and Torsten Hoefler. 2024. *Demystifying Chains, Trees, and Graphs of Thoughts.* arXiv:2401.14295 [cs] http://arxiv.org/abs/2401.14295
[6] Andrew Blair-Stanek, Nils Holzenberger, and Benjamin Van Durme. 2024. BLT: Can Large Language Models Handle Basic Legal Text?. In *Proceedings of the Natural Legal Language Processing Workshop 2024* (Miami, FL, USA, 2024-11), Nikolaos Aletras, Ilias Chalkidis, Leslie Barrett, Cătălina Goanță, Daniel Preoțiuc-Pietro, and Gerasimos Spanakis (Eds.). Association for Computational Linguistics, 216–232. doi:10.18653/v1/2024.nllp-1.18
[7] Andrew Blair-Stanek, Nils Holzenberger, and Benjamin Van Durme. 2023. Can GPT-3 Perform Statutory Reasoning?. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law* (Braga Portugal, 2023-06-19). ACM, 22–31. doi:10.1145/3594536.3595163
[8] Supreme Court California. 1993. Brown v. Poway Unified School Dist. 624 pages.
[9] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2023. *A Survey on Evaluation of Large Language Models.* arXiv:2307.03109 [cs] http://arxiv.org/abs/2307.03109
[10] Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. Benchmarking Large Language Models in Retrieval-Augmented Generation. *Proceedings of the AAAI Conference on Artificial Intelligence* 38, 16 (March 2024), 17754–17762. doi:10.1609/aaai.v38i16.29728
[11] Jonathan H. Choi and Daniel B. Schwarcz. 2023. AI Assistance in Legal Analysis: An Empirical Study. *SSRN Electronic Journal* (2023). doi:10.2139/ssrn.4539836
[12] Andrew Coan and Harry Surden. 2024. *Artificial Intelligence and Constitutional Interpretation.* doi:10.2139/ssrn.5018779
[13] Matthew Dahl, Varun Magesh, Mirac Suzgun, and Daniel E. Ho. 2024. Large Legal Fictions: Profiling Legal Hallucinations in Large Language Models. arXiv:2401.01301 [cs]
[14] Judy Davis. [n. d.]. LibGuides: Tort Law Research Guide: Practice Guides. https://lawlibguides.usc.edu/c.php?g=687841&p=4879061.
[15] Fabrizio Dell'Acqua, Edward McFowland, Ethan R. Mollick, Hila Lifshitz-Assaf, Katherine Kellogg, Saran Rajendran, Lisa Krayer, François Candelon, and Karim R. Lakhani. 2023. Navigating the Jagged Technological Frontier: Field Experimental Evidence of the Effects of AI on Knowledge Worker Productivity and Quality. (2023). doi:10.2139/ssrn.4573321
[16] James R. Dorsey, Bradley J. Gunn, Marc D. Simpson, Peter G. Mikhail, and Greta L. Bjerkness. 2023. Chapter 10. Eminent Domain. In *25 Minn. Prac., Real Estate Law.* Minnesota Practice Series, Vol. 25. Thomson West.
[17] Colin Doyle. 2024. *LLMs as Method Actors: A Model for Prompt Engineering and Architecture.* arXiv:2411.05778 http://arxiv.org/abs/2411.05778
[18] Lance Eliot. 2020. *AI and Legal Argumentation: Aligning the Autonomous Levels of AI Legal Reasoning.* arXiv:2009.11180 [cs] http://arxiv.org/abs/2009.11180
[19] Jamal Greene. 2011. The Rule of Law as a Law of Standards. 99 (2011).
[20] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948* (2025).
[21] Legal Information Institute. n.d.. res ipsa loquitur. https://www.law.cornell.edu/wex/res_ipsa_loquitur. Accessed: July 10th, 2024.
[22] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing*

Systems, Vol. 33. Curran Associates, Inc., 9459–9474.

[23] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050* (2023).

[24] Varun Magesh, Faiz Surani, Matthew Dahl, Mirac Suzgun, Christopher D. Manning, and Daniel E. Ho. 2024. Hallucination-Free? Assessing the Reliability of Leading AI Legal Research Tools. arXiv:2405.20362 [cs]

[25] Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2024. *GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models.* doi:10.48550/arXiv.2410.05229 arXiv:2410.05229 [cs]

[26] John J. Nay, David Karamardian, Sarah B. Lawsky, Wenting Tao, Meghana Bhat, Raghav Jain, Aaron Travis Lee, Jonathan H. Choi, and Jungo Kasai. 2023. *Large Language Models as Tax Attorneys: A Case Study in Legal Capabilities Emergence.* arXiv:2306.07075 [cs] http://arxiv.org/abs/2306.07075

[27] OpenAI. 2024. Learning to Reason with LLMs. https://openai.com/index/learning-to-reason-with-llms/ Accessed: 2025-02-03.

[28] OpenAI. 2025. OpenAI O1 system card. https://openai.com/index/openai-o1-system-card/. Accessed: January 31, 2025.

[29] 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 (2022), 27730–27744.

[30] Rohan Padhye. 2024. Software Engineering Methods for AI-Driven Deductive Legal Reasoning. In *Proceedings of the 2024 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software* (Pasadena CA USA, 2024-10-17). ACM, 85–95. doi:10.1145/3689492.3690050

[31] Robert Ramsey. 2024. Chapter 19. Criminal Justice Reform. In *Criminal Practice and Procedure*. New Jersey Practice Series, Vol. 31. Thomson West.

[32] Edwina L. Rissland. 1985. AI and Legal Reasoning Report. https://www.ijcai.org/Proceedings/85-2/Papers/111.pdf

[33] Antonin Scalia. 1989. The Rule of Law as a Law of Rules. 56, 4 (1989), 1175. doi:10.2307/1599672 jstor:1599672

[34] Michael Paul Thomas, Zaida Angulo McGhee, Brian D. Kahn, and Stacy L. La Scala. 2024. § 1:29. Presumption of Breach Arising from Type of Accident ("Res Ipsa Loquitur"). In *Torts (California Civil Practice)*. Bancroft Whitney.

[35] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. *arXiv preprint arXiv:1803.05355* (2018).

[36] Shubham Vatsal and Harsh Dubey. 2024. *A Survey of Prompt Engineering Methods in Large Language Models for Different NLP Tasks.* arXiv:2407.12994 [cs] http://arxiv.org/abs/2407.12994

[37] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models.* arXiv:2201.11903 [cs] http://arxiv.org/abs/2201.11903

[38] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903 [cs.CL] https://arxiv.org/abs/2201.11903