Measuring Faithfulness and Abstention: An Automated Pipeline for Evaluating LLM-Generated 3-ply Case-Based Legal Arguments

Li Zhang[1,*], Morgan Gray[1], Jaromir Savelka[2] and Kevin D. Ashley[1]

[1]Intelligent Systems Program, University of Pittsburgh, Pittsburgh, Pennsylvania, USA
[2]School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA

Abstract

Large Language Models (LLMs) demonstrate potential in complex legal tasks like argument generation, yet their reliability remains a concern. Building upon pilot work assessing LLM generation of 3-ply legal arguments using human evaluation, this paper introduces an automated pipeline to evaluate LLM performance on this task, specifically focusing on faithfulness (absence of hallucination), factor utilization, and appropriate abstention. We define hallucination as the generation of factors not present in the input case materials and abstention as the model's ability to refrain from generating arguments when instructed and no factual basis exists. Our automated method employs an external LLM to extract factors from generated arguments and compares them against the ground-truth factors provided in the input case triples (current case and two precedent cases). We evaluated eight distinct LLMs on three tests of increasing difficulty: 1) generating a standard 3-ply argument, 2) generating an argument with swapped precedent roles, and 3) recognizing the impossibility of argument generation due to lack of shared factors and abstaining. Our findings indicate that while current LLMs achieve high accuracy (over 90%) in avoiding hallucination on viable argument generation tests (Tests 1 & 2), they often fail to utilize the full set of relevant factors present in the cases. Critically, on the abstention test (Test 3), most models failed to follow instructions to stop, instead generating spurious arguments despite the lack of common factors. This automated pipeline provides a scalable method for assessing these crucial LLM behaviors, highlighting the need for improvements in factor utilization and robust abstention capabilities before reliable deployment in legal settings. Project page: Link.

Keywords

LLM Evaluation, Legal Argument Generation, Hallucination Measurement, Abstention, Trustworthy AI

1. Introduction

Large language models (LLMs) have demonstrated remarkable capabilities across various domains, including legal analysis and argumentation [1, 2, 3]. Their potential to streamline legal research, draft documents, and even generate arguments offers significant efficiency gains. However, their tendency to hallucinate facts or generate plausible but unsupported statements poses significant risks in legal applications, where accuracy and reliability are of utomost importance [4, 5]. Misguided decisions, ethical concerns, and even professional sanctions can result from relying on inaccurate AI-generated legal content [6, 7].

A critical challenge lies in ensuring the factual accuracy and appropriate reasoning behavior of LLMs when tasked with generating case-based legal arguments. Pilot work involving human evaluation of LLM-generated 3-ply arguments (plaintiff's argument citing precedent 1, defendant's counterargument distinguishing precedent 1 and citing precedent 2, plaintiff's rebuttal distinguishing precedent 2) indicated that while LLMs can produce structurally coherent arguments, their factual grounding and adherence to constraints can be problematic [3]. Specifically, LLMs may hallucinate, i.e., introduce

Proceedings of the Seventh International Workshop on Automated Semantic Analysis of Information in Legal Text (ASAIL 2025), 16 June 2025, Chicago, IL

*Corresponding author.
liz239@pitt.edu (L. Zhang); mag454@pitt.edu (M. Gray); jsavelka@andrew.cmu.edu (J. Savelka); ashley@pitt.edu (K. D. Ashley)
0000-0003-0375-1793 (L. Zhang); 0000-0002-3800-2103 (M. Gray); 0000-0002-3674-5456 (J. Savelka); 0000-0002-5535-0759 (K. D. Ashley)

factual elements (represented as 'factors' in case-based reasoning) not present in the source materials. Furthermore, they may fail to follow instructions appropriately, particularly negative constraints such as abstaining from generating an argument when the provided cases lack a sufficient factual basis for comparison. Existing evaluation methods often focus on general capabilities [8, 9, 10] but lack fine-grained metrics to assess these specific failure modes in the context of factor-based legal argumentation.

To address this gap, we introduce an automated pipeline for evaluating LLM performance in generating 3-ply, factor-based legal arguments. This pipeline specifically targets the assessment of hallucination, factor utilization (the extent to which relevant, available factors are used), and appropriate abstention. The core of our approach involves using an external LLM to analyze the arguments generated by the models under test, extracting the factors cited within them. These extracted factors are then compared against the ground-truth factors present in the input case materials to compute quantitative metrics for faithfulness and completeness.

The development of this automated evaluation pipeline enables a targeted assessment of LLM behavior in generating factor-based arguments. To guide this assessment, we pose the following research questions (RQs):

- **RQ1:** To what extent do LLMs exhibit measurable errors, specifically hallucination (citing non-existent factors) and incomplete factor utilization (omitting relevant available factors), when tasked with generating 3-ply case-based arguments from factor-represented inputs?
- **RQ2:** How effectively do LLMs adhere to instructions to abstain from argument generation when presented with input cases lacking common factors, and what is their propensity to generate spurious arguments under such conditions?
- **RQ3:** Can the proposed automated evaluation metrics effectively quantify distinct error types (hallucination, incomplete utilization, spurious generation) and successfully reveal performance variations across different LLMs and varying levels of task complexity?

The main contributions of this paper are: an automated evaluation pipeline specifically designed for assessing LLM-generated, factor-based legal arguments; novel metrics targeting hallucination, factor utilization, and abstention behavior in this context; an empirical evaluation of eight distinct LLMs (including open-source and proprietary models of varying sizes) on three argumentation tasks with increasing difficulty; and insights into the specific weaknesses of current LLMs regarding factual grounding and instruction following in legal argument generation.

The remainder of this paper is structured as follows: Section 2 discusses related work. Section 3 details the methodology of our automated evaluation pipeline. Section 4 describes the experimental setup, including the dataset, tasks, and models. Section 5 presents the results of our evaluation. Section 6 provides a qualitative error analysis. Section 7 concludes the paper. Finally, Section 8 acknowledges the limitations and suggests future work.

2. Background and Related Work

2.1. LLMs in the Legal Domain

Recent advances in LLMs, from open-source models such as Llama models [11] to proprietary systems such as GPT models [12], have demonstrated remarkable capabilities in natural language understanding and generation. This has spurred significant interest in their application within the legal domain, ranging from legal research assistance and contract analysis to case outcome prediction and automated legal document drafting [1]. While promising, the reliable deployment of these models requires careful consideration of their limitations, particularly concerning factual accuracy.

2.2. Computational Argumentation and Case-Based Reasoning

Computational models of legal argument, particularly those employing case-based reasoning (CBR), provide a foundation for analyzing and generating legal arguments. Early work pioneered the use of 'factors'—stereotypical fact patterns relevant to legal claims—in US trade secrets law, developing systems like HYPO that analyze and compare cases based on shared and distinguishing factors [13]. Subsequent systems like CATO introduced factor hierarchies [14], while others integrated rule-based and case-based approaches [15] or focused on predicting outcomes [16] and incorporating legal values [17]. Formal models of precedential constraint based on factors have also been developed [18]. Factors provide a structured representation suitable for evaluating the factual basis of arguments, as employed in this study.

2.3. Argument Generation with LLMs

Beyond general legal tasks, researchers are exploring the specific capability of LLMs to generate arguments. Some work has shown LLMs can assist humans in identifying legal factors [19, 20] or generate factor-based arguments in a structured manner [3]. However, the practical utility of such generated arguments hinges on their factual accuracy and logical coherence. This paper focuses on evaluating these aspects rather than proposing new generation methods.

2.4. Hallucination in LLMs

A significant challenge for LLMs is hallucination—the generation of content that is factually incorrect or unsupported by the provided input or established knowledge [5, 21]. Hallucinations can manifest as contradictions with the input prompt, conflicts with provided context, or deviations from real-world facts [22]. In high-stakes domains like law, where precision and truthfulness are critical [4], hallucination represents a major barrier to adoption. Mitigation strategies often involve techniques like chain-of-thought prompting [23] or retrieval-augmented generation (RAG) to ground responses in external sources [24]. Our work focuses on reliably detecting hallucination in the specific context of factor-based legal arguments.

2.5. Evaluation Metrics for Generated Text

Standard metrics for evaluating generated text, such as ROUGE [25], BLEU [26], and BERTScore [27], primarily measure surface-level similarity or semantic overlap with reference texts. While useful for tasks like summarization or translation, they are often insufficient for assessing factual accuracy, logical consistency, or adherence to constraints in complex generation tasks like legal argumentation [10]. Some legal benchmarks exist [8, 9], but metrics specifically tailored to evaluate faithfulness and abstention in factor-based reasoning remain underdeveloped. Our work aims to fill this gap by proposing automated metrics focused on these critical aspects.

2.6. Instruction Following in LLMs

The ability of LLMs to accurately follow complex instructions is crucial for their reliable use. Research has shown that while LLMs are increasingly capable of adhering to instructions, they can struggle with nuanced or complex constraints, particularly negative constraints (e.g., "do not generate X if condition Y is met") or situations requiring implicit recognition of task impossibility [28, 29]. Failure to follow instructions, such as the requirement to abstain from generating an argument when no factual basis exists, is one of the key failure modes investigated in this study.

3. Methodology

Our work employs a pipeline designed to automatically generate, assess, and score LLM performance on a structured legal argument generation task. This pipeline consists of several stages: scenario generation, argument generation by the models under test, automated factor extraction from the generated arguments, and quantitative scoring based on comparison with ground-truth inputs.

3.1. Task Definition: 3-Ply Argument Generation

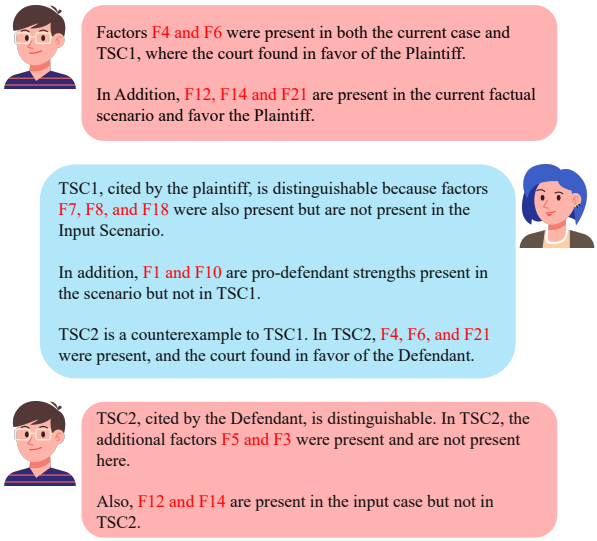

Figure 1: Three-ply Legal Argument Generation Scheme

The core task requires the LLM to generate a 3-ply legal argument within the U.S. trade secret law domain, following the structure established by Ashley [13]. Given a current fact situation (the 'current case') represented by factors, and two precedent trade secret misappropriation cases (TSC1 and TSC2), also represented by factors, the LLM must perform three steps: First, act as Plaintiff and argue for victory by citing TSC1 or TSC2 as an analogous precedent, focusing on shared pro-plaintiff factors. Second, act as Defendant, responding by distinguishing Plaintiff's cited case (highlighting differential factors) and citing the other precedent case as a counter-example favouring the defendant, focusing on shared pro-defendant factors. Third, act again as Plaintiff for a rebuttal, distinguishing the Defendant's cited case and emphasizing factors that differentiate the current case from the Defendant's precedent. This argumentative structure is illustrated in Figure 1.

3.2. Factor-Based Case Representation

Cases are represented using a standardized set of 26 legal factors pertinent to U.S. trade secret misappropriation law, derived from foundational work in legal AI [13, 30]. These factors encapsulate key factual aspects, such as circumstances surrounding disclosure, security measures implemented, characteristics of the information, and relevant employee conduct. Each factor is designated as typically favouring either the plaintiff (P) or the defendant (D). For instance, a case might be represented textually as:

```
[Case Name] [Outcome] [Factors: F1 Disclosure-in-negotiations (D), F4 Agreed-not-to-
    disclose (P), F6 Security-measures (P)]
```

This structured factor representation facilitates objective comparison between cases based on shared and distinguishing factors, providing the essential ground truth for our subsequent automated evaluation metrics.

3.3. Argument Generation and Evaluation Pipeline

The evaluation process follows a defined pipeline. First, legal scenarios (case triples) are generated according to specific criteria (detailed in Section 4.1). Second, each model under test receives these scenarios within a structured prompt (Section 4.2) and generates the 3-ply argument. Third, an automated process extracts the factors cited within the generated argument text (Section 3.4.1). Finally, these extracted factors are compared against the ground-truth factors from the input scenario to calculate performance metrics (Sections 3.4.2-3.4.4). The overall process is depicted in Figure 2.

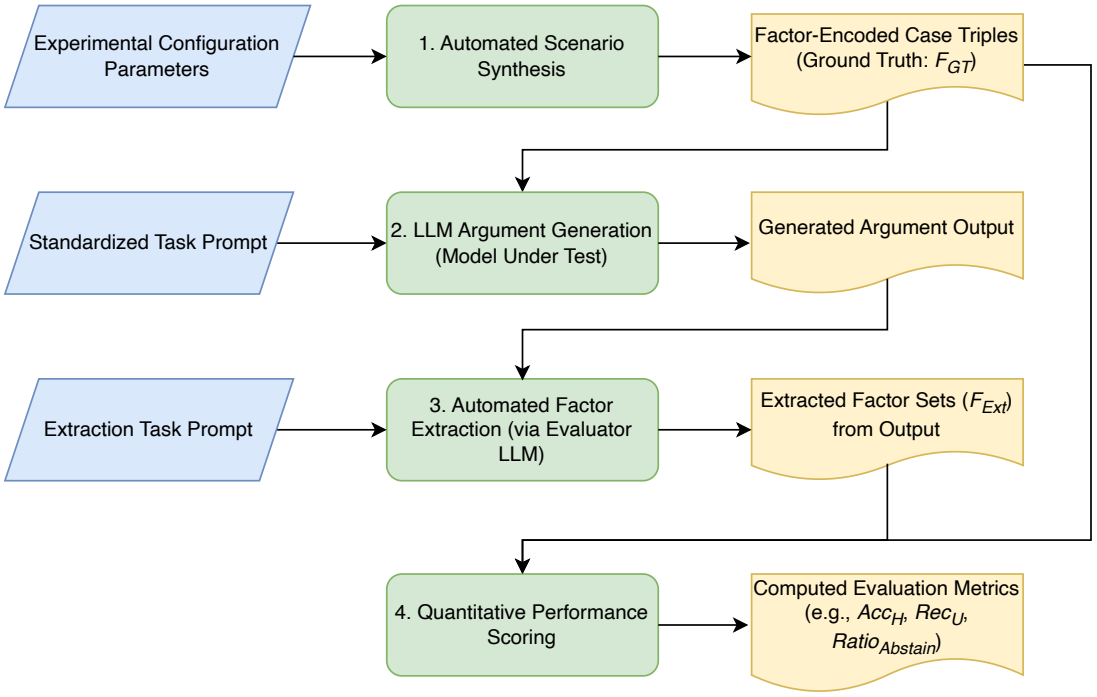

Figure 2: Overall Automated Evaluation Pipeline Flowchart

The core argument generation step takes the factor-represented case triple as input and invokes the chosen model with specific parameters (Section 4.4) to produce the 3-ply textual argument. The LLM output, along with metadata, is logged for subsequent analysis.

3.4. Automated Metrics Definition

To evaluate the generated 3-ply arguments quantitatively, we developed automated metrics focused on faithfulness (absence of hallucination), factor utilization (completeness), and adherence to abstention instructions. These metrics rely on comparing the factors cited within the generated argument against the ground-truth factors present in the input cases.

3.4.1 Factor Extraction: An external, high-capability LLM (`GPT-4.1`) serves as an automated evaluator. For each generated 3-ply argument produced by a model under test, this evaluator LLM is prompted to analyze the argument text. Its task is to identify and extract the specific sets of factors that the model under test asserted existing in each case in the triples (Current Case - CC, TSC1, TSC2). Let $F_{Ext,c}$ be the set of factors extracted by the evaluator for case $c \in \{CC, TSC1, TSC2\}$. Similarly, let $F_{GT,c}$ be the set of actual ground-truth factors present in the input for case c.

3.4.2 Hallucination Metric: Hallucination is operationally defined as the assertion by the model under test that a specific factor exists in a specific case when, according to the ground-truth input for that case, it is not present. We quantify this by summing the hallucinations across all three cases and normalizing by the total number of ground-truth factors in the input triple. The *Hallucination Accuracy* (Acc_H) is calculated as:

$$Acc_H = \left(1 - \frac{N_H}{N_{GT}}\right) \times 100\%$$

Here, N_H is the total count of hallucinated factors across the three cases:

$$N_H = \sum_{c \in \{CC, TSC1, TSC2\}} |\{f \in F_{Ext,c} \mid f \notin F_{GT,c}\}|$$

And N_{GT} represents the total count of factors across the three ground-truth input cases (sum of factors per case, not unique factors):

$$N_{GT} = \sum_{c \in \{CC, TSC1, TSC2\}} |F_{GT,c}|$$

A higher Acc_H indicates greater faithfulness, meaning the argument relies less on unsupported factual assertions specific to each case.

3.4.3 Factor Utilization Metric: Factor utilization assesses how comprehensively the model under test mentions the available ground-truth factors for the specific cases they belong to. We compute *Factor Utilization Recall* (Rec_U) by summing the correctly identified factors for each case and normalizing by the total number of ground-truth factors.

$$Rec_U = \left(\frac{N_U}{N_{GT}} \right) \times 100\%$$

where N_U is the total count of utilized ground-truth factors correctly mentioned for their respective cases across the triple:

$$N_U = \sum_{c \in \{CC, TSC1, TSC2\}} |F_{Ext,c} \cap F_{GT,c}|$$

N_{GT} is defined as above. A higher Rec_U indicates that the generated argument incorporates more of the factual elements provided in the input, correctly associating them with their respective cases.

3.4.4 Abstention Metric: Test 3 specifically tests the model's ability to abstain when argument generation is impossible due to a lack of shared factors. The primary measure for this test is the *Abstention Ratio* ($Ratio_{Abstain}$). Let N_{SA} be the number of successfully executed abstentions and N_{TA} be the total number of test triples requiring abstention. The ratio is calculated as:

$$Ratio_{Abstain} = \left(\frac{N_{SA}}{N_{TA}} \right) \times 100\%$$

A higher $Ratio_{Abstain}$ indicates better adherence to instructions to abstain. For this test, we also report Hallucination Accuracy (Acc_H) to characterize the nature of the arguments generated when models failed to abstain (Section 5.3).

3.5. Rationale for External LLM-based Evaluation

Employing a highly capable LLM (`GPT-4.1`) for the factor extraction step (Section 3.4.1) provides substantial advantages in scalability and consistency compared to manual annotation across potentially hundreds or thousands of generated arguments. Although the evaluator LLM is not infallible, our spot checks indicated high accuracy in identifying factor mentions within the generated text structures. This automated approach enables large-scale, reproducible evaluation across numerous models and experimental conditions. Potential limitations associated with this method are acknowledged in Section 8.

4. Experimental Design

4.1. Dataset Generation and Structure

The dataset used in this study was synthetically generated using a custom tool designed to create controlled case triples for evaluating specific argumentation phenomena within the U.S. trade secret domain. Each generated triple includes a factor-represented current case, a potential plaintiff precedent (TSC1), and a potential defendant counter-precedent (TSC2).

The generation process allows for specifying several parameters, including the number of cases, the complexity level which controlls the number of factors per case, typically ranging from complexity-1 to complexity+1, and, crucially, the scenario 'mode'. We generated data across three distinct modes, each designed to test different facets of LLM reasoning and instruction following:

- **Arguable Sets:** These triples contain sufficient overlapping factors between the current case and the respective precedents (TSC1 for plaintiff, TSC2 for defendant), with aligned outcomes, facilitating standard 3-ply argument generation. These are used for Test 1.
- **Reordered Sets:** In these triples, common factors exist, but the typical roles based on outcomes are reversed (TSC1 favors Defendant, TSC2 favors Plaintiff). These are used for Test 2, primarily testing robustness to the reordered precedent cases compared to Test 1.
- **Non-arguable Sets:** These triples are constructed such that there are no common factors between the current case and either TSC1 or TSC2. These are specifically designed for Test 3 to evaluate the model's ability to recognize the impossibility of argument generation and abstain as instructed.

For this study, we generated sets of 30 triples with complexity of 12, for each of the 'Arguable', 'Reordered', and 'Non-arguable' modes, resulting in a total dataset of 90 triples used across the three experimental tests. Example case structures for each mode are illustrated in Table 1. This structured dataset allows for targeted testing of baseline argument generation (Arguable), adherence to specific instructions under potentially confusing conditions (Reordered/Swapped Roles), and the crucial ability to recognize factual impossibility and follow abstention instructions (Non-arguable).

Table 1
Examples of Different Dataset Scenario Modes

Mode	Current Case	TSC1	TSC2
Arguable	F4: Agreed-not-to-disclose (P)* F5 Agreement-not-specific (D)† F23 Waiver-of-confidentiality (D)	outcome: Plaintiff F2 Bribe-employee (P) F4: Agreed-not-to-disclose (P)* F16 Info-reverse-engineerable (D)	outcome: Defendant F2 Bribe-employee (P) F5: Agreement-not-specific (D)† F12: Outsider-disclosures-restricted (P)
Reordered	F4: Agreed-not-to-disclose (P)† F5 Agreement-not-specific (D)* F23 Waiver-of-confidentiality (D)	outcome: Defendant F2 Bribe-employee (P) F5: Agreement-not-specific (D)* F12: Outsider-disclosures-restricted (P)	outcome: Plaintiff F2 Bribe-employee (P) F4: Agreed-not-to-disclose (P)† F16 Info-reverse-engineerable (D)
Non-arguable	F6: Security-measures (P) F22: Invasive-techniques (P)	outcome: Plaintiff F1: Disclosure-in-negotiations (D) F27: Disclosure-in-public-forum (D)	outcome: Defendant F16: Info-reverse-engineerable (D) F24: Info-obtainable-elsewhere (D)

Notes: Common factors between Current Case and TSC1 marked with *; common factors between Current Case and TSC2 marked with †.

4.2. Core Prompt Structure

For each test, the LLMs under evaluation were provided with a structured prompt designed to give sufficient context and clear instructions. The prompt included a description of the 3-ply argument test (as the scheme shown in Figure 1), relevant background on trade secret misappropriation law, and the factor representations for the specific input cases (current case, TSC1, TSC2). Crucially, the prompt also contained explicit instructions regarding the desired output format and an abstention condition: it stated that if no common factors could be found to support an analogy for a given ply, the model should output a specific phrase (e.g., "Cannot generate argument due to lack of common factors") and stop processing that ply, rather than fabricating an argument. This abstention instruction was particularly relevant for evaluating performance on Test 3. An example of the full prompt structure is provided in Appendix A and B.

4.3. Curriculum for Testing

We evaluated the selected LLMs on three distinct tests, leveraging the different modes of our generated dataset:

- **Test 1 on Arguable Sets: Standard Argument Generation.** Using the 'Arguable' case triples, models generated the standard 3-ply argument (Plaintiff cites TSC1, Defendant cites TSC2, Plaintiff rebuts TSC2). This test assesses baseline performance regarding hallucination and factor utilization when arguments are factually supported.
- **Test 2 on Reordered Sets: Swapped Precedent Roles.** Also using the 'Arguable' triples, this test required models to perform the 3-ply argument but with the order of TSC1 and TSC2 swapped (TSC1 favoring the Defendant, TSC2 favoring the Plaintiff). Critically, models were not given the precedent name to cite; instead, they had to select the appropriate precedent (TSC1 or TSC2) by analyzing which case's outcome supported their argumentative goal.
- **Test 3 on Non-arguable Sets: Abstention Test.** Utilizing the 'Non-arguable' case triples, models were prompted to generate the standard 3-ply argument. The expected correct behavior, however, was for the model to recognize the lack of common factors required for analogical reasoning and follow the explicit instruction to abstain from generating spurious arguments for triple. This test directly probes the ability to identify task impossibility and adhere to negative constraints.

These tests present progressively challenging scenarios designed to probe different aspects of LLM reliability in legal argument generation.

4.4. Models Evaluated

We selected eight distinct LLMs, representing a range of sizes, architectures, and access types (open-source and commercial). The models evaluated were:

- *GPT-4o* (OpenAI)
- *GPT-4o-mini* (OpenAI)
- *Llama-3-70B-8192* (Meta)
- *Llama-3-8B-8192* (Meta)
- *Llama-4-Maverick-17B-128e-instruct* (Meta)
- *Llama-4-Scout-17B-16e-instruct* (Meta)
- *DeepSeek-R1-Distill-Llama-70B* (DeepSeek)
- *Qwen-QWQ-32B* (Alibaba)

This selection aimed to cover a spectrum of capabilities, including models ranging from comparatively small (Llama-3-8B) to large (GPT-4o, Llama-3-70B), proprietary and open-source options, Mixture-of-Experts architectures (Llama-4 variants), and models optimized for reasoning tasks (Qwen-QWQ-32B, DeepSeek-R1-Distill-Llama-70B). These models were accessed via their respective APIs at the time of experimentation (May 2025).

4.5. Implementation Details

To ensure comparability across models and tests, consistent generation parameters were employed for all LLM invocations within the argument generation pipeline. We used a `temperature` setting of 0 since the task is focused on expected correct output. A `max_tokens` limit of 500 was set, which proved sufficient for the typical length of a 3-ply argument structure (for reasoning models, the limit was set as 5,000). Other standard parameters included `top_p`=1, `frequency_penalty`=0, and `presence_penalty`=0. The factor extraction step performed by the external evaluator LLM (`GPT-4.1`) also utilized fixed deterministic settings to ensure consistency in the evaluation process itself.

5. Results

We analyze the performance of the eight evaluated LLMs across the three distinct tests using the automated metrics for Hallucination Accuracy (Acc_H, Section 3.4.2), Factor Utilization Recall (Rec_U, Section 3.4.3), and Abstention Ratio ($Ratio_{Abstain}$, Section 3.4.4), as defined previously. The results are presented separately for each test type: Test 1 (Arguable), Test 2 (Reordered), and Test 3 (Non-arguable).

Table 2
LLM Hallucination Accuracy (Acc_H, % Across Tests)

Model	Test 1 (Arguable)	Test 2 (Reordered)	Test 3 (Non-arguable)
Llama-3-8B-8192	92.39	93.53	84.91
Llama-3-70B-8192	96.36	97.83	91.55
Llama-4-Scout-17B-16e-instruct	96.45	97.21	86.69
Llama-4-Maverick-17B-128e-instruct	96.96	**98.15**	94.35
GPT-4o-mini	96.95	96.79	88.42
GPT-4o	**99.64**	97.36	**99.16**
DeepSeek-R1-Distill-Llama-70B	91.38	90.26	88.94
Qwen-QWQ-32B	91.73	90.83	90.08

Table 3
LLM Factor Utilization Recall (Rec_U, % Across Tests 1 & 2)

Model	Test 1 (Arguable)	Test 2 (Reordered)
Llama-3-8B-8192	64.63	61.42
Llama-3-70B-8192	77.51	74.62
Llama-4-Scout-17B-16e-instruct	69.47	63.77
Llama-4-Maverick-17B-128e-instruct	53.95	51.40
GPT-4o-mini	42.47	49.55
GPT-4o	**85.22**	**76.61**
DeepSeek-R1-Distill-Llama-70B	72.61	69.58
Qwen-QWQ-32B	61.40	52.51

Table 4
LLM Abstention Ratio ($Ratio_{Abstain}$, % for Test 3)

Model	Abstention Ratio (Test 3)
Llama-3-8B-8192	0.00
Llama-3-70B-8192	3.33
Llama-4-Scout-17B-16e-instruct	0.00
Llama-4-Maverick-17B-128e-instruct	50.00
GPT-4o-mini	0.00
GPT-4o	**86.67**
DeepSeek-R1-Distill-Llama-70B	23.33
Qwen-QWQ-32B	56.67

5.1. Hallucination Accuracy Results

As shown in Table 2, the Hallucination Accuracy (Acc_H) is generally very high for Tests 1 and 2 across most models. These tests involve generating arguments based on provided 'Arguable' scenarios, either in a standard format (Test 1) or with swapped precedent roles (Test 2). GPT-4o achieves near-perfect

accuracy (>99% for Test 1, >97% for Test 2), indicating exceptional faithfulness in citing only those factors genuinely shared between the relevant cases in these scenarios. Other models, including GPT-4o-mini, Llama-3-70B, and the Llama-4 variants, also demonstrate high accuracy, typically above 96% on these tests. This suggests that when instructed to generate arguments based on factor representations where supporting shared factors exist, current leading LLMs are capable of doing so with minimal hallucination according to our metric. Smaller or specialized models like DeepSeek, Qwen, and Llama-3-8B show slightly lower but still respectable accuracy, generally above 90%.

For Test 3 (Non-arguable), where no shared factors exist and models were instructed to abstain, the Acc_H remains surprisingly high for several models, particularly GPT-4o (99.16%) and Llama-4-Maverick (94.35%). This high accuracy, however, must be interpreted cautiously alongside the primary goal of abstention and the recall results (Section 5.3).

5.2. Factor Utilization Recall Results

Factor Utilization Recall (Rec_U), presented in Table 3, measures the completeness of the arguments by quantifying the proportion of available supporting factors that were correctly identified and used by the LLM for Tests 1 and 2. The results reveal a more varied picture than accuracy. For Tests 1 and 2, GPT-4o again leads, utilizing over 85% of available factors in the standard test (Test 1) and over 76% in the swapped-role test (Test 2). Llama-3-70B also performs strongly, achieving recall scores above 74% for both tests. Other models exhibit a wider range of performance. For instance, GPT-4o-mini shows significantly lower recall (around 42-50%), suggesting it generates arguments that are factually accurate (high Acc_H) but lack comprehensiveness. The Llama-4 variants, DeepSeek, Qwen, and Llama-3-8B fall between these extremes, with recall generally ranging from 50% to 70% on the arguable tests. This indicates that while models can avoid making up facts, they often fail to incorporate the full set of relevant facts available in the input materials into their generated arguments.

5.3. Abstention Test Performance

Test 3 was designed specifically to test the models' ability to follow the instruction to abstain when faced with 'Non-arguable' scenarios lacking shared factors. The primary metric for this test is the Abstention Ratio ($Ratio_{Abstain}$), presented in Table 4. We also consider Hallucination Accuracy (Acc_H) from Table 2 for Test 3 to understand the nature of arguments generated when models failed to abstain.

As shown in Table 4, the ability to correctly abstain varies significantly across models. GPT-4o achieved the highest abstention ratio (86.67%), successfully following the instruction in the majority of non-arguable cases. Qwen-QWQ-32B (56.67%) and Llama-4-Maverick (50.00%) also demonstrated some capability to abstain. However, several models, including Llama-3-8B, Llama-4-Scout, and GPT-4o-mini, had an abstention ratio of 0.00%, indicating they failed to abstain in any of the test instances. Llama-3-70B also performed poorly with a very low abstention ratio (3.33%).

For models that failed to abstain and instead generated spurious arguments, their Hallucination Accuracy (Acc_H) on Test 3 (Table 2) is informative. For example, GPT-4o, even when it rarely failed to abstain, maintained very high Acc_H (99.16%), meaning its spurious arguments were largely free of hallucinated factors not in the input cases. Llama-4-Maverick also showed high Acc_H (94.35%) in such instances. This suggests that their failure was primarily in not following the abstention instruction, rather than fabricating factors. Other models that failed to abstain also generally maintained relatively high Acc_H (mostly above 84%), indicating that the spurious arguments, while incorrect, were mostly based on factors present in the input cases rather than completely fabricated information.

Overall, this critical test of instruction following reveals a significant weakness in most LLMs. The inability to reliably recognize task impossibility and adhere to negative constraints is a major concern for their deployment in sensitive applications. Even models that performed well on argument generation (Tests 1 & 2) struggled significantly with abstention.

5.4. Comparative Analysis

Across Tests 1 and 2 (argument generation), GPT-4o demonstrates the strongest performance, achieving the highest Hallucination Accuracy and leading significantly in Factor Utilization Recall, suggesting its arguments are both faithful and comprehensive. Llama-3-70B generally ranks second, showing strong accuracy and good recall. Llama-4-Maverick also performs well in terms of accuracy on these tests, though its recall is moderate.

The performance gap between the top models (GPT-4o, Llama-3-70B) and others is more pronounced in Factor Utilization Recall than in Hallucination Accuracy for Tests 1 and 2. Models like GPT-4o-mini, Llama-3-8B, and Qwen often achieve reasonable accuracy but struggle with recall, producing less complete arguments. The Llama-4 variants and DeepSeek fall in the mid-range for both metrics on these arguable tests.

Test 3 (Abstention) reveals a critical dimension of model capability. GPT-4o stands out with the highest Abstention Ratio (Table 4), indicating a superior ability to follow instructions to abstain. Qwen-QWQ-32B and Llama-4-Maverick show moderate success in abstention, while several models, including some high performers on Tests 1 & 2 like Llama-3-70B, almost completely failed to abstain. This highlights that strong performance on generative tasks does not necessarily translate to robust instruction following for negative constraints. The failure to abstain, even when explicitly instructed, is a significant concern. The failure modes on Test 3 varied, but very few models performed the task as intended by consistently and correctly abstaining.

6. Error Analysis

To gain deeper insights into the quantitative results and understand the nature of the errors identified by our automated metrics, we conducted an error analysis. We selected LLM outputs for manual review primarily based on instances where the automated metrics indicated significant deviations from desired performance. This included cases with lower Hallucination Accuracy ($Acc_H < 95\%$), notably low Factor Utilization Recall (Rec_U), outputs from models exhibiting generally weaker performance on a specific test, and, critically, all instances where models failed to produce the correct abstention output on Test 3. The analysis involved a manual review of the selected LLM-generated argument texts. Each generated argument was compared against the ground-truth factors provided in the corresponding input case triple (Current Case, TSC1, TSC2) and the specific instructions given in the core prompt (including the 3-ply structure requirements and the abstention rule). During the review, observed errors were categorized into distinct types related to hallucination, incomplete factor utilization, and failures in following instructions, particularly regarding the abstention task.

Hallucination Errors (Primarily Tests 1 & 2): Although quantitative results showed high Acc_H, qualitative analysis identified infrequent instances, primarily in lower-performing models.

- *Factor Misattribution:* Citing a factor as present in one case (e.g., the Current Case) when it actually belonged to a different case in the input triple (e.g., TSC1).

Factor Misattribution: GPT-4o

"Plaintiff's Argument: ... Factors F3 ... F21 Knew-info-confidential (P), F23 Waiver-of-confidentiality (D), and F25 Info-reverse-engineered (D) were present in both the input case and TSC1 ..."

For example, in one instance of a GPT-4o output (Box 1), factor F21 was incorrectly attributed to TSC1 by the model, as it was not present in the ground-truth factors for TSC1.

Incomplete Factor Utilization Errors (Tests 1 & 2): This was a more common issue across models, reflected in the Rec_U scores.

- *Omission of Shared Factors:* Failing to identify or mention relevant factors that were shared between the Current Case and the precedent being cited (TSC1 for Plaintiff's first ply, TSC2 for Defendant's ply).

- *Omission of Distinguishing Factors:* Failing to identify or mention factors that differentiate the Current Case from the precedent being discussed, particularly when the task required distinguishing (Defendant's ply distinguishing TSC1, Plaintiff's rebuttal distinguishing TSC2).

Omission of Shared Factors & Omission of Distinguishing Factors: Llama-3-8B-8192

"Plaintiff's Argument: ... Factors F4 Agreed-not-to-disclose (P) and F6 Security-measures (P) were present in both the input case and TSC1 ..."
"Plaintiff's Rebuttal: ... In TSC2, the additional factors ... F25 Info-reverse-engineered (D), F27 Disclosure-in-public-forum (D) were present and are not present in input case."

For instance, with Llama-3-8B-8192 (Box 2), the model failed to mention that F7 Brought-tools (P) was a shared pro-plaintiff factor in both the current case and TSC1 during the Plaintiff's Argument. Additionally, in the Plaintiff's Rebuttal, the model did not point out that F12 Outsider-disclosures-restricted (P) was a distinguishing factor present in the current case but not in TSC2.

Instruction Following / Abstention Errors (Test 3): This was the most significant failure mode observed across nearly all models.

- *Failure to Abstain:* The most common error was generating a spurious 3-ply argument structure despite the lack of common factors and the explicit instruction to output a specific abstention phrase. We observe that even in instances where the model does not successfully abstain, the resulting legal arguments do not exhibit factual misattribution or hallucination.
- *Incorrect Abstention Phrase:* In rare cases where a model attempted to signal impossibility, it failed to use the exact required phrase ("No common factor between the current case and the TSC1/TSC2"), using a variation instead.
- *Spurious Reasoning within Failed Abstention:* When models incorrectly generated arguments, the reasoning was inherently flawed. This often involved discussing factors from the cases as if they provided a basis for comparison, even though no common factors existed, or fabricating connections.

Failure to Abstain: GPT-4o

"Plaintiff's Argument: ... F22 Invasive-techniques (P), F26 Deception (P) were present in the input case and support the Plaintiff's position ..."

This output (Box 3) illustrates GPT-4o failing to follow the instruction to abstain when presented with a non-arguable scenario in Test 3. Instead of outputting the specified abstention phrase, the model proceeded to generate an argument. Notably, while the generation of any argument was incorrect in this context, the factors cited in this whole output were present in the input case material, indicating a failure in instruction adherence rather than a hallucination errors.

Spurious Reasoning within Failed Abstention: Deepseek-R1-Distill-Llama-70B

"<think> ... The first step is to check if there are common factors between the input current case and either TSC1 or TSC2. If there are none, I should stop ... <\think>"
"Plaintiff's Argument: Factors F6 Security-measures (P) and F21 Knew-info-confidential (P) were present in both the input case and TSC1 ..."

As shown in the example (Box 4), even when a model's reasoning trace explicitly mentioned the condition for stopping, it might still proceed to generate a spurious argument, failing to follow the critical abstention instruction. Additionally, the model claimed that there were common factors between the input case and TSC1.

This qualitative analysis complements the quantitative metrics, highlighting that even with high accuracy, completeness remains a challenge, and adherence to negative constraints like abstention is a critical weakness for current LLMs in this legal argument generation context.

7. Conclusion

This paper introduced and applied an automated pipeline to evaluate the performance of eight LLMs on generating 3-ply, factor-based legal arguments, focusing specifically on faithfulness, completeness, and the ability to follow abstention instructions. Our evaluation, guided by three research questions, yielded the following conclusions:

Regarding **RQ1** (hallucination and incomplete factor utilization), our results show that while most evaluated LLMs exhibit high Hallucination Accuracy ($Acc_H > 90 - 95\%$ in Tests 1 & 2), indicating they generally avoid citing non-existent factors when generating arguments in viable scenarios, they struggle with completeness. Factor Utilization Recall (Rec_U) varied significantly (from ≈40% to ≈85% in Tests 1 & 2), demonstrating that LLMs often omit relevant, available factors from the input cases, leading to potentially superficial arguments.

Concerning **RQ2** (adherence to abstention instructions), the evaluation revealed a critical weakness across almost all models. When presented with non-arguable scenarios (Test 3) and explicitly instructed to abstain, most models failed to follow this directive, as measured by our Abstention Ratio (Table 4). Instead of abstaining, they generated spurious arguments. Only a few models showed a significant ability to abstain, with GPT-4o performing best, yet still not perfectly. This highlights a fundamental inability in most current LLMs to reliably recognize task impossibility and follow negative constraints.

Addressing **RQ3** (effectiveness of automated metrics), the proposed metrics (Acc_H, Rec_U, and $Ratio_{Abstain}$) successfully quantified distinct error types. Acc_H effectively measured faithfulness (absence of hallucination). Rec_U captured incomplete factor utilization in argument generation tasks (Tests 1 & 2). $Ratio_{Abstain}$ directly measured the critical capability of adherence to abstention instructions in Test 3. Together, these metrics clearly revealed performance variations across different LLMs and task complexities, demonstrating the pipeline's utility in diagnosing specific weaknesses.

In summary, while LLMs show promise in generating factually grounded components of legal arguments based on structured inputs, significant improvements are needed in ensuring comprehensive reasoning (completeness) and, most crucially, in robust instruction following, particularly regarding negative constraints and the ability to abstain appropriately. These deficiencies must be addressed before LLMs can be reliably deployed for substantive legal argumentation tasks.

8. Limitations and Future Work

This study is subject to several limitations. The evaluation utilizes synthetic, factor-represented cases, simplifying the nuances of real-world legal texts and reasoning. The accuracy of our automated metrics inherently depends on the performance of the external LLM used for factor extraction, introducing a potential layer of error. The specific operationalization of our metrics, particularly for the abstention test, could be further refined. Furthermore, the findings are based on a specific dataset size, prompt structure, and set of LLMs, potentially limiting the generalizability of the precise quantitative results, although the qualitative trends are likely indicative.

Future research should aim to address these limitations. Evaluating performance on larger, more diverse datasets, including those derived from real-world legal documents (which would necessitate robust factor extraction from text as a preliminary step [31]), is a crucial next step. Further validation and refinement of the automated metrics, potentially including comparisons with human expert judgments on argument quality beyond factor usage, would strengthen the evaluation pipeline. Investigating the underlying reasons for the observed deficiencies in recall and abstention through model interpretability or targeted probing could inform the development of more reliable models. Finally, exploring novel prompting strategies, fine-tuning approaches, or architectural modifications specifically designed to enhance completeness and robust instruction adherence in legal argument generation remains a vital avenue for future work.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] F. Ariai, G. Demartini, Natural language processing for the legal domain: A survey of tasks, datasets, models, and challenges, arXiv preprint arXiv:2410.21306 (2024).

[2] D. M. Katz, D. Hartung, L. Gerlach, A. Jana, M. J. Bommarito II, Natural language processing in the legal domain, arXiv preprint arXiv:2302.12039 (2023).

[3] M. A. Gray, L. Zhang, K. D. Ashley, Generating case-based legal arguments with llms, in: Proceedings of the 4th ACM Computers and Law Symposium, 2025.

[4] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al., On the opportunities and risks of foundation models, arXiv preprint arXiv:2108.07258 (2021).

[5] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, P. Fung, Survey of hallucination in natural language generation, ACM Computing Surveys 55 (2023) 1–38.

[6] D. U. S. de la Osa, N. Remolina, Artificial intelligence at the bench: Legal and ethical challenges of informing—or misinforming—judicial decision-making through generative ai, Data & Policy 6 (2024) e59.

[7] J. J. Avery, P. S. Abril, A. del Riego, Chatgpt, esq.: Recasting unauthorized practice of law in the era of generative ai, Yale JL & Tech. 26 (2023) 64.

[8] N. Guha, J. Nyarko, D. Ho, C. Ré, A. Chilton, A. Chohlas-Wood, A. Peters, B. Waldon, D. Rockmore, D. Zambrano, et al., Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models, Advances in Neural Information Processing Systems 36 (2024).

[9] H. Li, Y. Chen, Q. Ai, Y. Wu, R. Zhang, Y. Liu, Lexeval: A comprehensive chinese legal benchmark for evaluating large language models, arXiv preprint arXiv:2409.20288 (2024).

[10] Z. Fei, X. Shen, D. Zhu, F. Zhou, Z. Han, S. Zhang, K. Chen, Z. Shen, J. Ge, Lawbench: Benchmarking legal knowledge of large language models, arXiv preprint arXiv:2309.16289 (2023).

[11] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., Llama: Open and efficient foundation language models, arXiv preprint arXiv:2302.13971 (2023).

[12] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al., Gpt-4 technical report, arXiv preprint arXiv:2303.08774 (2023).

[13] K. D. Ashley, Modeling Legal Argument: Reasoning with Cases and Hypotheticals, The MIT Press, Cambridge, MA, 1990.

[14] V. Aleven, K. D. Ashley, Teaching case-based argumentation through a model and examples: Empirical evaluation of an intelligent learning environment, in: Artificial intelligence in education, volume 39, Citeseer, 1997, pp. 87–94.

[15] E. L. Rissland, D. B. Skalak, Cabaret:rule interpretation in a hybrid architecture, International Journal of Man-machine Studies 34 (1991) 839–887.

[16] S. Brüninghaus, K. D. Ashley, Predicting outcomes of case based legal arguments, in: Proceedings of the 9th International conference on Artificial Intelligence and Law, ACM, 2003, pp. 233–242.

[17] M. Grabmair, Predicting trade secret case outcomes using argument schemes and learned quantitative value effect tradeoffs, in: Proceedings of the 16th ICAIL, 2017, pp. 89–98.

[18] J. F. Horty, Modifying precedential constraint, Journal of Artificial Intelligence Law 30 (2021) 1–24.

[19] M. A. Gray, J. Savelka, W. M. Oliver, K. D. Ashley, Empirical legal analysis simplified: reducing complexity through automatic identification and evaluation of legally relevant factors, Philosophical Transactions of the Royal Society A 382 (2024) 20230155.

[20] J. Savelka, K. D. Ashley, The unreasonable effectiveness of large language models in zero-shot semantic annotation of legal texts, Frontiers in Artificial Intelligence 6 (2023) 1279794.
[21] Y. A. Yadkori, I. Kuzborskij, A. György, C. Szepesvári, To believe or not to believe your llm, arXiv preprint arXiv:2406.02543 (2024).
[22] Y. Zhang, Y. Li, L. Cui, D. Cai, L. Liu, T. Fu, X. Huang, E. Zhao, Y. Zhang, Y. Chen, et al., Siren's song in the ai ocean: a survey on hallucination in large language models, arXiv preprint arXiv:2309.01219 (2023).
[23] S. Dhuliawala, M. Komeili, J. Xu, R. Raileanu, X. Li, A. Celikyilmaz, J. Weston, Chain-of-verification reduces hallucination in large language models, arXiv preprint arXiv:2309.11495 (2023).
[24] T. Vu, M. Iyyer, X. Wang, N. Constant, J. Wei, J. Wei, C. Tar, Y.-H. Sung, D. Zhou, Q. Le, et al., Freshllms: Refreshing large language models with search engine augmentation, arXiv preprint arXiv:2310.03214 (2023).
[25] C.-Y. Lin, Rouge: A package for automatic evaluation of summaries, in: Text summarization branches out, 2004, pp. 74–81.
[26] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, Bleu: a method for automatic evaluation of machine translation, in: Proceedings of the 40th annual meeting of the Association for Computational Linguistics, 2002, pp. 311–318.
[27] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi, Bertscore: Evaluating text generation with bert, arXiv preprint arXiv:1904.09675 (2019).
[28] B. Wen, J. Yao, S. Feng, C. Xu, Y. Tsvetkov, B. Howe, L. L. Wang, Know your limits: A survey of abstention in large language models, arXiv preprint arXiv:2407.18418 (2024).
[29] S. Feng, W. Shi, Y. Wang, W. Ding, V. Balachandran, Y. Tsvetkov, Don't hallucinate, abstain: Identifying llm knowledge gaps via multi-llm collaboration, arXiv preprint arXiv:2402.00367 (2024).
[30] K. D. Ashley, Artificial intelligence and legal analytics: new tools for law practice in the digital age, Cambridge University Press, 2017.
[31] M. Gray, J. Savelka, W. Oliver, K. Ashley, Automatic identification and empirical analysis of legally relevant factors, in: Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law, 2023, pp. 101–110.

A. Prompt Structure for 3-Ply Argument Generation

The following shows the structure of the prompt provided to the LLMs for the 3-ply argument generation task.

Example Prompt

TASK

In this task, we will formulate legal arguments based on trade secret misappropriation claims using a structured approach. Follow the steps outlined below for consistency and clarity.

Legal Problem Context

In this problem, we aim to develop arguments using factors critical to trade secret misappropriation claims. Typically, the Plaintiff alleges that the Defendant has misappropriated their trade secret. For instance, Kentucky Fried Chicken (KFC) could claim misappropriation if an employee disclosed their secret recipe, which is a blend of herbs and spices, by publishing it in a cookbook.

Factors may support either the Plaintiff (P) or the Defendant (D). The Plaintiff might emphasize measures they took to protect the recipe, while the Defendant could argue that the recipe was already disclosed to outsiders. Based on the factors provided, construct a three-part argument as detailed below.

Instructions

1. **IMPORTANT:** If there is no common factor between the current case and the TSC1/TSC2, you need to say "No common factor between the input current case and the TSC1/TSC2" and stop generating any argument.
2. **Construct a 3-Ply Argument:**
    a) **Plaintiff's Argument:** Present an argument in favor of the Plaintiff's position by i. citing a relevant Trade Secret Case (TSC1/TSC2) with a similar favorable outcome; ii. Highlighting shared factors between the input current case and the TSC1/TSC2.
    b) **Defendant's Counterargument:** Refute the Plaintiff's position by i. Distinguishing the cited TSC1/TSC2 based on differing factors; ii. Citing a counterexample (a TSC1/TSC2 with a Defendant-favorable outcome) and drawing an analogy to the input current case.
    c) **Plaintiff's Rebuttal:** Address and distinguish the counterexample, reinforcing the Plaintiff's original argument.
3. Use Provided Factors: Base your arguments on the factors outlined, ensuring logical consistency.

Example Input and Output

Example Current Case

- F1 Disclosure-in-negotiations (D)
- F4 Agreed-not-to-disclose (P)
- F6 Security-measures (P)
- F10 Secrets-disclosed-outsiders (D)
- F12 Outsider-disclosures-restricted (P)
- F14 Restricted-materials-used (P)
- F21 Knew-info-confidential (P)

Example TSC1 outcome Plaintiff

- F4 Agreed-not-to-disclose (P)

- F6 Security-measures (P)
- F7 Brought-tools (P)
- F8 Competitive-advantage (P)
- F18 Identical-products (P)

Example TSC2 outcome Defendant

- F3 Employee-sole-developer (D)
- F4 Agreed-not-to-disclose (P)
- F5 Agreement-not-specific (D)
- F6 Security-measures (P)
- F21 Knew-info-confidential (P)

Example Output (json format)

Plaintiff's Argument: Factors F4 Agreed-not-to-disclose (P) and F6 Security-measures (P) were present in both the current case and TSC1, where the court found in favor of the Plaintiff. In Addition, Factors F12 Outsider-disclosures-restricted (P), F14 Restricted-materials-used (P), F21 Knew-info-confidential (P) are present in the current case and favor the Plaintiff.

Defendant's Counterargument: TSC1, cited by the plaintiff is distinguishable because factors F7 Brought-tools (P), F8 Competitive-advantage (P), and F18 Identical-products (P) were also present, but are not present in the current case. In addition, F1 Disclosure-in-negotiations (D) and F10 Secrets-disclosed-outsiders (D) are pro-defendant strengths present in the current case but not in TSC1. TSC2 is a counterexample to TSC1. In TSC2, F4 Agreed-not-to-disclose (P), F6 Security-measures (P), and F21 Knew-info-confidential (P) were present in both the current case and TSC2 and the court found in favor of the Defendant.

Plaintiff's Rebuttal: TSC2, cited by the Defendant is distinguishable. In TSC2, the additional factors F5 Agreement-not-specific (D) and F3 Employee-sole-developer (D) were present and are not present in the current case. Also, F12 Outsider-disclosures-restricted (P) and F14 Restricted-materials-used (P) are present in the current case but not in TSC2.

Current Case, TSC1, and TSC2 ...

B. Prompt Structure for Factor Extraction

The following shows the structure of the prompt provided to the LLM (`GPT-4.1`) for the factor extraction task.

Example Prompt

TASK

You are tasked with extracting factors from the 3-ply argument.

Example Input (json format)

Plaintiff's Argument: Factors F4 Agreed-not-to-disclose (P) and F6 Security-measures (P) were present in both the current case and TSC1, where the court found in favor of the Plaintiff.

In Addition, Factors F12 Outsider-disclosures-restricted (P), F14 Restricted-materials-used (P), F21 Knew-info-confidential (P) are present in the current case and favor the Plaintiff.

Defendant's Counterargument: TSC1, cited by the plaintiff is distinguishable because factors F7 Brought-tools (P), F8 Competitive-advantage (P), and F18 Identical-products (P) were also present, but are not present in the current case. In addition, F1 Disclosure-in-negotiations (D) and F10 Secrets-disclosed-outsiders (D) are pro-defendant strengths present in the current case but not in TSC1. TSC2 is a counterexample to TSC1. In TSC2, F4 Agreed-not-to-disclose (P), F6 Security-measures (P), and F21 Knew-info-confidential (P) were present in both the current case and TSC2 and the court found in favor of the Defendant.

Plaintiff's Rebuttal: TSC2, cited by the Defendant is distinguishable. In TSC2, the additional factors F5 Agreement-not-specific (D) and F3 Employee-sole-developer (D) were present and are not present in the current case. Also, F12 Outsider-disclosures-restricted (P) and F14 Restricted-materials-used (P) are present in the current case but not in TSC2.

Example Output (json format)
Current Case

- F1 Disclosure-in-negotiations (D)
- F4 Agreed-not-to-disclose (P)
- F6 Security-measures (P)
- F10 Secrets-disclosed-outsiders (D)
- F12 Outsider-disclosures-restricted (P)
- F14 Restricted-materials-used (P)
- F21 Knew-info-confidential (P)

TSC1

- F4 Agreed-not-to-disclose (P)
- F6 Security-measures (P)
- F7 Brought-tools (P)
- F8 Competitive-advantage (P)
- F18 Identical-products (P)

TSC2

- F3 Employee-sole-developer (D)
- F4 Agreed-not-to-disclose (P)
- F5 Agreement-not-specific (D)
- F6 Security-measures (P)
- F21 Knew-info-confidential (P)

3-Ply Argument to be Extracted ...