CaseFacts: A Benchmark for Legal Fact-Checking and Precedent Retrieval

Akshith Reddy Putta[*]**, Jacob Devasier, Chengkai Li**
University of Texas at Arlington
{akshith.putta, cli}@uta.edu, jacob.devasier@mavs.uta.edu

Abstract

Automated Fact-Checking has largely focused on verifying general knowledge against static corpora, overlooking high-stakes domains like law where truth is evolving and technically complex. We introduce **CaseFacts**, a benchmark for verifying colloquial legal claims against U.S. Supreme Court precedents. Unlike existing resources that map formal texts to formal texts, CaseFacts challenges systems to bridge the semantic gap between layperson assertions and technical jurisprudence while accounting for temporal validity. The dataset consists of 6,294 claims categorized as SUPPORTED, REFUTED, or OVERRULED. We construct this benchmark using a multi-stage pipeline that leverages Large Language Models (LLMs) to synthesize claims from expert case summaries, employing a novel semantic similarity heuristic to efficiently identify and verify complex legal overrulings. Experiments with state-of-the-art LLMs reveal that the task remains challenging; notably, augmenting models with unrestricted web search degrades performance compared to closed-book baselines due to the retrieval of noisy, non-authoritative precedents. We release CaseFacts[1] to spur research into legal fact verification systems.

1 Introduction

Automated Fact-Checking (AFC) has emerged as a critical safeguard against misinformation, yet existing systems primarily operate within general-knowledge domains, validating claims against corpora like Wikipedia (Aly et al., 2021; Akhtar et al., 2022). However, high-stakes domains such as law present unique challenges that render general-purpose AFC systems inadequate. In the legal sphere, verification requires more than surface-level textual overlap; it demands the retrieval of authoritative precedent, the interpretation of complex holdings, and the temporal awareness to determine if a ruling is still valid.

The primary bottleneck in Legal AFC is the semantic gap between public discourse and judicial writing. While most people assert claims in colloquial language (e.g., "The Supreme Court banned school prayer"), the supporting evidence exists in formal, technical jurisprudence (e.g., "state officials may not compose an official state prayer"). Current legal benchmarks, such as LegalBench (Guha et al., 2023) and CaseHOLD (Zheng et al., 2021), facilitate legal reasoning and judgment prediction but operate strictly within the formal register—mapping legalese to legalese. They do not evaluate a model's ability to bridge the linguistic disparity between an informal claim and a formal ruling (Chen et al., 2025; Mori et al., 2025).

Furthermore, legal truth is temporally fragile. A claim that was true in 1990 may be false today if the underlying precedent has been overruled. Existing datasets rarely explicitly model this validity constraint, often treating legal documents as static text rather than a dynamic system of evolving rules.

To address these limitations, we introduce **CaseFacts**, a new benchmark dataset designed to evaluate the verification of colloquial claims against U.S. Supreme Court (SCOTUS) rulings. Unlike previous resources, CaseFacts is constructed to test three specific dimensions of legal verification: (1) Retrieval across the semantic gap, mapping informal assertions to technical case law; (2) Reasoning against hard negatives, distinguishing true principles from plausible but factually refuted distortions; and (3) Precedent changes over time, identifying when a claim relies on a precedent that has been explicitly OVERRULED.

Constructing such a dataset presents significant challenges due to the scarcity of expert-annotated training data. We introduce a rigorous, multi-stage pipeline leveraging Large Language Mod-

[*] The author is a UTA affiliate and attends Coppell High School.

[1] https://github.com/idirlab/supreme-court-dataset

els (LLMs) to synthesize claims from expert summaries of 3,299 SCOTUS cases sourced from Oyez[2]. Our pipeline generates diverse claim-evidence pairs and employs a novel LLM-as-a-Judge verification loop to filter for factuality, resolve internal contradictions, and ensure stylistic diversity. To validate the OVERRULED class, we implement a heuristic filtering method based on semantic similarity intervals to efficiently locate inter-case contradictions, which are then verified against case metadata. The final dataset consists of 6,294 claims, including a high-quality, human-annotated test set.

We benchmark state-of-the-art LLMs and retrieval models on CaseFacts, yielding several critical insights. First, we find that allowing LLMs unrestricted web search does not necessarily improve performance; in our baselines, a search-enabled GPT-4o performed worse than a naive, closed-book setting, often retrieving noisy or tangentially relevant cases. Second, we demonstrate that while standard Natural Language Inference (NLI) models fail to grasp the nuances of legal overrulings, embedding models fine-tuned on our training set significantly improve retrieval recall.

In summary, our contributions are as follows:

- We introduce **CaseFacts**, the first large-scale benchmark for verifying colloquial legal claims against SCOTUS precedents, featuring distinct SUPPORTED, REFUTED, and OVERRULED classes.

- We propose a robust synthetic data generation pipeline that resolves intra-case and inter-case inconsistencies, using semantic similarity heuristics to efficiently identify legal overrulings and confirmations in large corpora.

- We provide comprehensive benchmarks demonstrating that CaseFacts poses a significant challenge to current LLMs, particularly in bridging the semantic gap and retrieving the precise controlling authority required for legal verification.

2 CaseFacts Benchmark

2.1 Task Formulation

We formulate legal fact-checking as a combined retrieval and verification task. Given a colloquial legal claim c and a knowledge base of case law $\mathcal{K}$, the system must output a verdict $y \in \{\textsc{Supported}, \textsc{Refuted}, \textsc{Overruled}\}$ and a set of supporting evidence SCOTUS cases $E \subset \mathcal{K}$. A claim is SUPPORTED if it is entailed by the holding of a valid case in $\mathcal{K}$. It is REFUTED if it contradicts the holding of a valid case. It is OVERRULED if it relies on a holding from a case $k_{old} \in \mathcal{K}$ that has been explicitly overturned by a subsequent reversing case $k_{new} \in \mathcal{K}$.

2.2 Data Source

Our primary source for legal documentation is Oyez, a multimedia archive of the US Supreme Court. We utilized the “facts,” “question,” and “conclusion” fields for each case, each of which are summaries synthesized by legal experts using original Supreme Court documentation. We refer to these fields as case “evidence”. These expert summaries provide a high-quality, structured foundation for generating synthetic legal claims. We filtered the corpus to include only cases containing all three fields, resulting in a collection of 3,299 cases (as of July 2025).

2.3 Dataset Creation Pipeline

We implemented a multi-stage pipeline to generate synthetic claims and ensure their factuality, consistency, and diversity. To mitigate self-preference bias during validation (Wataoka et al., 2025), we utilized different model families for generation (Qwen3-Next-80B-A3B-Thinking) (Team, 2025) and post-generation validation (gemma-3-27b-it) (Team et al., 2025). We use open-weight models only to promote reproducibility of our dataset creation method. All LLMs’ temperature was set to 0.6 and top-p to 0.9.

2.3.1 Initial Claim Generation

We prompted the generation model with the case evidence from each source case and instructed it to generate claims derived from the case’s legal principles (Prompt 1). To ensure the retrieval task remained non-trivial and generalizable, we explicitly constrained the model to exclude case-specific identifiers, such as party names or dates. Furthermore, we enforced strict stylistic constraints prohibiting “court-report phrasing” (e.g., “The court found that...”). This ensures the output consists of direct, atomic legal assertions rather than summaries of judicial opinions. This process yielded an initial pool of 10,471 unique claims.

[2] https://www.oyez.org/

2.3.2 Intra-Case Validation

We applied filters to ensure factual consistency and minimize redundancy among claims derived from the same case. These inconsistencies would hinder the creation of a high-quality dataset.

Factuality Check We employed an LLM-as-a-judge to verify that each claim was strictly entailed by the source case evidence (Prompt 2). We also explicitly instructed the model to adhere to a closed-world assumption, prohibiting reliance on outside legal knowledge or assumptions not present in the provided case summary. This process removed 1,484 claims that failed to meet the expected entailment criteria, resulting in a pool of 8,987 factually consistent claims.

Claim Inconsistencies An issue that was noticed was the presence of the LLM creating factually-inconsistent claims for claims within the same case. These are errors associated with LLM generation. We refer to these as "contradictions," even though most of them are smaller factual inconsistencies with each other when relating to the evidence of the case. Another issue that was noticed was the presence of very similar, redundant claims generated from the same case, and we refer to these as "overlaps." This is possibly due to the LLM generating unnecessary claims, although it has been instructed not to. As the aim of the dataset is to establish generate claims which each have a "legal principle" from the case. Therefore, redundant claims must be removed to maintain a dataset consisting of high-quality, meaningfully-different claims. We collectively refer to overlaps and contradictions as claim inconsistencies.

Intra-Case Inconsistency Resolution We detected 90 pairs of internally contradictory claims and 1,568 pairs of semantically redundant claims using Prompts 3 and 4, for pairs within the same case. The smaller number of contradictory claims is possibly due to the factuality check removing most of the inconsistent claims, with the remaining identified with this pass. A resolution step (Prompts 5, 6) removed one claim from each of the contradictory and redundant pairs. This removed 1,404 claims, resulting in a filtered pool of 7,583 supported claims.

2.3.3 Inter-Case Validation

We addressed inconsistencies across different cases to identify cross-case confirmations (claims supported by the rulings of multiple cases) and construct the OVERRULED class with the detected contradictions. A naive approach would involve checking all possible cross-case pairs (N^2) for inconsistencies, but this requires over 25 million comparisons, which is computationally infeasible for LLMs, even smaller ones like gemma3-27b-it.

Inadequacy of Standard NLI To filter these pairs efficiently, we first attempted to use a standard Natural Language Inference (NLI) model (`textattack/bert-base-uncased-MNLI`) to identify contradictions and entailments across the full dataset. However, qualitative analysis revealed that standard NLI models consistently failed to grasp the deeper legal entailment required to identify overrulings and confirmations, rendering this approach unusable.

Similarity-Based Filtering Consequently, we adopted a heuristic approach based on semantic similarity (detailed in Section 3.2). We analyzed the distribution of inconsistent pairs and observed that 95% of contradictions and overlaps occur within specific semantic similarity intervals (0.70–0.95 for contradictions; 0.80–0.97 for overlaps, as detailed in Table 4). These ranges contained 199,203 pairs for contradictions and 15,874 pairs for overlaps. We therefore restricted the expensive LLM-based verification to pairs falling within these high-risk intervals, reducing the workload to a manageable subset while maintaining high recall.

Identifying Overrulings We identified 1,946 contradicting pairs, 0.98% of the sampled range. In the legal domain, such contradictions often signify that one case overrules another. We resolved these using Prompt 9, which provided the model with the ruling dates of both cases to determine temporal precedence. To minimize false positives, the prompt instructed the model to favor an interpretation of consistency where possible, reserving the OVERRULED label for clear legal overrulings. We then validated potential overrulings against Oyez disposition metadata. If the overruling case had a valid disposition (e.g., "reversed" or "remanded"), we labeled the claim from the older case as OVERRULED and appended the newer case to its evidence. This process established the OVERRULED class, a distinct contribution of our benchmark.

Identifying Confirmations For the overlap detection task, we used prompt 8, yielding 12.31% of the pairs, 1,955 samples. Prompt 10 was used for resolving the overlapping claim pairs. For the overlapping claim pairs, we have updated the ground truth cases of the remaining claim to consist of both cases from the claim pair that overlapped, as the claim is confirmed by multiple cases. If there is a conflict when merging ground truth cases, where one claim is to merge with or merge under other claims, we select the pair with the highest similarity and instruct the LLM to indicate if the other claim can be merged under this pair. If so, the 3rd claim's case will be included as part of the original claim pair's cases.

2.3.4 Negative Claim Generation (Refuted)

To create the REFUTED class, we generated negations of the valid SUPPORTED claims using a two-step process designed to produce non-trivial false claims that sound like plausible legal principles. OVERRULED claims are by nature false and cannot have negations.

Step 1: Plausible Negation We first prompted the model (Prompt 11) to generate a plausible but factually incorrect negation of the original claim based on the case facts. This ensured the negation was grounded in the case context rather than being a generic or trivial contradiction, e.g., simply placing *not* in the claim.

Step 2: Principle Generalization Initial negations were often lengthy (avg. 212 chars) and contained specific conditional clauses (e.g., "unless X occurs..."), acting as artifacts that could allow models to distinguish false claims by length or specificity alone. To address this, we applied a refinement LLM pass (Prompt 12) instructing the model to rewrite the false claim as a concise, general legal principle. The model was explicitly instructed to remove specific case details and unconditional qualifiers while preserving the (false) core legal assertion. This reduced the average length to 108 chars (see Figure 1) and ensured the REFUTED claims matched the stylistic distribution of the SUPPORTED claims.

2.4 Human Verification

The test set consists of 500 claims verified by 2 human annotators. Annotators were instructed to select clear, factual claims free of identifying information. For OVERRULED claims and claims with multiple ground truth cases, annotators verified the verdict against the full set of evidence. From a pool of 638 annotated claims, 574 were validated as high-quality, from which we sampled the final 500. Table 1 highlights that the OVERRULED class had a lower human-verification pass rate compared to other classes, indicating the inherent complexity of validating legal overrulings.

Category	**Selected/Total Claims**	**Percentage**
Overall	574/638	89.96%
Supported	353/384	91.93%
Refuted	177/195	90.77%
Overruled	44/59	74.58%

Table 1: Quality assessment statistics for the human-annotated test set. Percentages denote the proportion of claims selected as high-quality samples within each category.

2.5 Dataset Statistics

Dataset	Total	Supported	Refuted	Overruled
Train set	5,794	2,605	2,732	457
Test set (500)	500	280	177	43

Table 2: Overview of the dataset statistics.

The final **CaseFacts** benchmark consists of a training set of 5,794 claims and a test set of 500 claims. The class distributions are detailed in Table 2. Each sample shares a unique `fact_id` with its corresponding negation and we ensured that no `fact_id` overlaps between the training and test sets and between claims and their negations, to prevent data leakage. The claims were randomly sampled when selecting whether to use the SUPPORTED claim or its REFUTED negation.

In addition to the claims, we release the 3,299 SCOTUS cases used for dataset creation and which serve as ground truth evidence for the claims.

2.6 Evaluation Metrics

We evaluate system performance using a tiered framework designed to assess both retrieval quality and decision-making reliability. All metrics are macro-averaged across claims.

Assessing Retrieval Quality To evaluate the quality of retrieved case law, we utilize the EVIDENCE SCORE. This metric is calculated as the recall score between the predicted and gold case sets. If a system fails to retrieve at least half of the gold supporting cases within the top five results (Recall@5 < 0.5), the Evidence Score is penalized

EvaluationJudgement		Count	Percentage
Standard	Consistent	8987	85.83%
	Inconsistent	1483	14.16%
Naive	Consistent	5964	56.96%
	Inconsistent	491	4.69%
	Error	4016	38.35%

Table 3: Comparison of standard and naive factuality evaluation results, over 10,471 claims, showing counts and proportions of factuality judgment outcomes.

to zero. We introduce this threshold because legal research that misses the majority of controlling precedents is functionally useless to a user, regardless of how precise the other retrieved documents might be. This ensures that high scores reflect a robust retrieval of the core legal principles rather than incidental hits.

Assessing Verdict Reliability While we report standard VERDICT ACCURACY (a binary measure of whether the predicted verdict matches the gold label), our primary metric for system ranking is the VERDICT SCORE. This joint, un-weighted metric is computed as the product of the Evidence Score and the Verdict Accuracy.

We prioritize the Verdict Score because it rigorously penalizes ungrounded correctness—instances where a model guesses the correct verdict but cites irrelevant or incorrect cases as evidence. In the high-stakes legal domain, a correct answer derived from faulty reasoning is as dangerous as an incorrect answer. By requiring sufficiently accurate retrieval as a prerequisite for a positive score, this metric ensures that systems are only rewarded when they are right for the right reasons. The k=5 recall also prevents fact-checking systems from using random, incidental case hits as evidence.

3 Experiments

3.1 Naive Factuality Check

When performing the regular factuality check in the process of creating the dataset, we have also ran an experiment where the evidence was not provided to the LLM. We used prompt 13, and the results are displayed in Table 3. We can observe from this that the LLM is more hesitant to process when evidence is not provided, from the high number of error cases: occurring when the LLM doesn't give the output in the requested format. This is possible due to the LLM's hesitance when instructed to make a decision based on its internal knowledge, when its internal knowledge is not sufficient. In contrast, the LLM never gave outputs in an erroneous format when provided the case's evidence, even though the prompts (2 and 13) are quite similar.

3.2 Similarity Distribution of Cross-Case Claim Inconsistencies

To locate contradictions and overlapping claims across cases with our compute resource limits, we stratified the space of cross-case claim pairs by their semantic similarity and inspected a uniform subsample from each bin. First, we computed the semantic similarity scores of all possible claim pair combinations, using Qwen3-Embedding-8B (Zhang et al., 2025b).

We then performed stratified sampling to estimate the density of inconsistencies across the similarity spectrum. We drew a uniform sample of 1,000 pairs from each 0.1 similarity interval between 0.2 and 0.9. The highest interval (0.9–1.0) contained fewer than 1,000 total pairs; consequently, all pairs in this range were included. Conversely, the 0.0–0.2 interval contained negligible data and was excluded as the pairs were semantically unrelated. For each sampled pair, we employed an LLM-as-a-judge (Prompts 7 and 8) to detect contradictions and overlaps, utilizing the full evidence from both source cases.

Results We then inspected the subsample predictions and computed percentile statistics of similarity for contradiction and overlap detections. Both contradictions and overlaps are highly concentrated toward the upper end of the similarity scale: contradictions cluster in the high-similarity region (roughly $\geq$0.7) and overlaps are concentrated even further toward the top (roughly $\geq$0.8), as can be observed in Table 4. This is inherently logical, as semantically similar claims are more likely to cover similar legal principles.

Overlaps exhibit a substantially higher positive-rate within their window than contradictions, while contradictions are rarer. This necessitated the additional resolution step with the filter by disposition for the contradictions. These results justify restricting the LLM-judge to a high-similarity slice of the pairwise space: doing so covers the large majority of contradictions and overlaps while keeping compute cost manageable.

Category	Percentile	Similarity Range
Contradictions	50%	[0.8203, 0.9180]
Contradictions	68%	[0.8086, 0.9220]
Contradictions	75%	[0.8052, 0.9258]
Contradictions	95%	[0.7032, 0.9491]
Overlaps	50%	[0.9023, 0.9336]
Overlaps	68%	[0.8658, 0.9428]
Overlaps	75%	[0.8516, 0.9453]
Overlaps	95%	[0.8047, 0.9688]

Table 4: Statistics for contradictions and overlaps, including counts and similarity score distributions. The 95% percentile range was chosen for addressing cross-claim claim inconsistencies: contradictions and overlaps.

3.3 Baseline Experiments

We evaluate two LLM baselines for Supreme Court claim verification that differ only in their access to external information at inference time. Both baselines use the same prompt (14), evaluation, and LLM. GPT-4o was selected as the LLM to avoid self-preference bias, as it is not part of the Qwen3 or Gemma3 model families (Wataoka et al., 2025).

The prompt presents the model with a legal claim and instructs it to determine the verdict against U.S. Supreme Court case law, with the provided verdict definitions. In addition to predicting a verdict, the model is required to identify the Supreme Court cases that justify the decision and to rank them by importance. To ensure a balanced evaluation, the prompt includes an explicit list of valid Supreme Court case names and instructs the model to cite only from this list. This list consists of the 3,299 cases collected from Oyez that were used to create the dataset.

Baseline Setting There are two settings for the baseline system: Naive (no-search), and Search enabled. Both baselines were evaluated on the 3 metrics described in Section 2.6. In the naive setting, the model receives only the prompt. The model must rely solely on its internal knowledge to determine the correct verdict and select the cases to be used as evidence. This baseline evaluates the model's ability to perform reasoning and case recall without external evidence retrieval. In the search setting, the model is additionally allowed to perform web search during inference, using OpenAI's web search tool[3]. This enables the model to consult external sources when performing the fact-checking task.

Model	Metric	Performance
Naive	Evidence Score	**0.309**
Naive	Verdict Accuracy	**0.684**
Naive	Verdict Score	**0.260**
Search	Evidence Score	0.281
Search	Verdict Accuracy	0.656
Search	Verdict Score	0.228

Table 5: Comparison of evidence and verdict metrics for LLM configurations with and without search.

Results From Table 5, it is evident that the major challenge for this benchmark dataset is gathering evidence, as the evidence score (case recall metric) is much lower than the verdict accuracy. This points to the verdicts being easier to predict by the LLM, as both search baselines perform similarly on the verdict prediction. However, for a fact-checking application, the quality of the evidence retrieved is quite important for users' trustworthiness (Anand et al., 2022; Schlichtkrull et al., 2023), hence why we use the composite metric of "verdict score" as our primary metric for this dataset.

The Naive baseline outperforms the Search baseline on all three metrics, as can be observed from Table 5. This pattern indicates that allowing unrestricted web search did not improve, and in fact degraded, both evidence retrieval quality and the joint evidence-weighted verdict performance. A primary contributor is that web search sometimes surfaced cases outside the 3,299 case list or returned noisy, tangentially relevant cases, which led to lower overlap with the gold supporting cases therefore penalties by the evaluation metrics. By contrast, the Naive setting, relying on the model's internal knowledge and the constrained case list, produced more consistent case selections and higher overall scores, although not by much. These results suggest that any retrieval augmentation should be tightly constrained to the validated case set and paired with stronger re-ranking or filtering.

3.4 Finetuning Semantic Similarity Models

In addition to prompting-based LLM baselines, we finetune dense embedding models for map-

[3] https://platform.openai.com/docs/guides/tools-web-search

Model	Training Epochs	Recall@1	Recall@5	Recall@10
bge-base-en-v1.5	1	0.3978 → 0.4348 (+0.0370)	0.5964 → 0.6858 (+0.0895)	0.6709 → 0.7649 (+0.0940)
	3	0.3978 → 0.5372 (+0.1394)	0.5964 → 0.7456 (+0.1492)	0.6709 → 0.8043 (+0.1333)
	5	0.3978 → 0.5458 (+0.1480)	0.5964 → 0.7615 (+0.1652)	0.6709 → 0.8279 (+0.1570)
Qwen3-Embedding-0.6B	1	**0.4480 → 0.6440 (+0.1960)**	0.7288 → 0.8533 (+0.1245)	0.8089 → 0.8993 (+0.0904)
	3	0.4480 → 0.6342 (+0.1862)	0.7288 → 0.8524 (+0.1235)	0.8089 → 0.9114 (+0.1025)
MiniLM-L6-v2	1	0.2820 → 0.3544 (+0.0724)	0.5146 → 0.5592 (+0.0447)	0.6036 → 0.6719 (+0.0683)
	3	0.2820 → 0.3975 (+0.1155)	0.5146 → 0.6704 (+0.1558)	0.6036 → 0.7450 (+0.1414)
	5	0.2820 → 0.4316 (+0.1496)	**0.5146 → 0.6878 (+0.1733)**	**0.6036 → 0.7821 (+0.1785)**

Table 6: Recall@k comparisons (base → trained) for embedding models across different training epochs. Values in parentheses are improvements from trained to base. The greatest performance increase in each column is bolded.

ping legal claims to the relevant SCOTUS cases, with CaseFacts' training set. This experiment isolates the retrieval component of the broader fact-checking task and assesses whether supervised representation learning improves the ability to identify legally relevant precedents.

Given a legal claim from CaseFacts as a query, the model must retrieve the Supreme Court cases that are the ground truth evidence for that claim. Each case is represented by its textual description, consisting of the case facts, questions and conclusions. Performance is measured using the test set, and the metrics are Recall@k for $k \in \{1, 5, 10\}$.

We evaluated three base encoder families (BGE (Xiao et al., 2023), Qwen3-Embedding-0.6B (Zhang et al., 2025b), and MiniLM (Wang et al., 2020)) and compared each base model to versions fine-tuned with MultipleNegativesRankingLoss for 1, 3, and 5 epochs. Fine-tuning encourages the model to embed claims closer to their corresponding cases while pushing apart unrelated cases, aligning the embedding space with the structure of the dataset.

Results Table 6 reports that, as expected, across all models and epochs, supervised fine-tuning yields substantial improvements over the corresponding pretrained baselines. Fine-tuning consistently improves Recall@1, indicating that training substantially increases the likelihood that the most relevant Supreme Court case is ranked first. This is particularly pronounced for stronger and larger base models: Qwen3-Embedding-0.6B shows the largest single-step improvement at Recall@1 (+19.6 points after one epoch), while bge-base-en-v1.5 and MiniLM-L6-v2 exhibit smaller, but steady gains as training epochs increase. This pattern of increasing gains for more epochs holds for the smaller models, but Qwen3-Embedding-0.6B's performance decreases from the 1 to 3 epochs, leading us to not train and evaluate Qwen3-Embedding-0.6B for 5 epochs.

This demonstrates the usefulness of CaseFacts' training set as data-augmentation for the retrieval portion of fact-checking, displaying that trained retrieval models can be complementary to an LLM-based fact-checking system. This approach can be an alternative to naive and web-search LLM fact-checkers in the legal domain.

4 Related Work

Legal Reasoning and Benchmarks. Recent efforts have established robust benchmarks to evaluate Large Language Models (LLMs) on legal tasks. LegalBench (Guha et al., 2023) provides a comprehensive suite of tasks ranging from clause classification to rule application. While LegalBench covers various aspects of legal reasoning, none of its constituent tasks directly tackle the problem of automated fact-checking, where a system must verify an external claim against a corpus. Our work can be viewed as a synthesis of multiple LegalBench reasoning skills—requiring both the retrieval of relevant precedent and the entailment capabilities necessary to verify a claim.

Similarly, CaseHOLD (Zheng et al., 2021) focuses on identifying case holdings—the governing legal rules applied to specific facts that serve as binding precedent. However, this benchmark operates strictly within the formal legal register, using judicial citations as prompts rather than colloquial claims. Additionally, it frames the task as multiple-choice matching where distractors are simply irrelevant precedents selected via TF-IDF, rather than the adversarial distortions found in real-world misinformation. Our benchmark addresses these gaps by verifying informal assertions against formal rulings, testing resilience against subtle misinterpretations rather than just retrieval precision.

The Semantic Gap in Legal Retrieval A primary challenge in legal fact-checking is the linguistic disparity between colloquial claims and formal legal texts. Chen et al. (2025) and Mori et al. (2025) investigate this friction, highlighting the difficulty of retrieving relevant legal provisions using informal or layperson queries. This issue is further compounded by the structural complexity of legal documents; Reuter et al. (2025) identify that lexical redundancy and fragmented information in legal datasets cause standard RAG systems to fail frequently. Similarly, Ajay Mukund and Easwarakumar (2025) argue that traditional semantic similarity metrics often miss the nuanced relevance of statutes, requiring dynamic adaptation rather than static retrieval. The limited performance observed in these studies underscores the difficulty of the task and motivates our focus on bridging the gap between informal political discourse and the technical language of SCOTUS opinions.

Domain-Specific Fact-Checking There is a growing paradigm shift in automated fact-checking, moving away from "one-size-fits-all" systems (Putta et al., 2025; Braun et al., 2024) toward domain-specific architectures. Wang et al. (2025) and Devasier et al. (2025) argue that specialized domains—such as congressional law or medicine—require distinct pipelines that general-purpose systems cannot support. Hu et al. (2025) recently addressed this in the context of Legal Question Answering by fine-tuning models to reduce hallucinations; however, their work focuses on generating truthful answers to questions, whereas our work focuses on the task of verifying external claims against precedents.

To address the scarcity of real-world training data in these specialized domains, recent works have successfully employed synthetic data generation. Zhao and Flanigan (2025) and Zhang et al. (2025a) demonstrate that creating synthetic factual claims from source texts can significantly improve performance on downstream verification tasks. This methodology is supported by broader surveys in the field, such as Nadăş et al. (2025), which validate the use of LLMs to generate high-quality training data in domains constrained by data scarcity and privacy. Following this methodology, we leverage LLMs to generate diverse claim-evidence pairs.

5 Conclusion

In this work, we introduced **CaseFacts**, a benchmark designed to address the unique challenges of automated fact-checking in the legal domain. By bridging the semantic gap between colloquial assertions and formal Supreme Court precedents, CaseFacts moves beyond traditional "legalese-to-legalese" tasks, offering a more realistic evaluation of how legal information is consumed in public discourse and social media. A key contribution of our work is the rigorous modeling of precedent validity through the OVERRULED class. By developing a scalable pipeline to identify and verify overrulings, we demonstrate that legal truth is not static and that robust verification systems must account for the dynamic evolution of case law and the formation of new legal precedent.

Our experiments reveal significant limitations in current state-of-the-art models. We observed that providing LLMs with unrestricted web search often degrades performance due to the retrieval of noisy or irrelevant case law, highlighting the need for retrieval systems that prioritize authoritative and precise evidence. Furthermore, while fine-tuning embedding models on our synthetic training data yielded substantial gains in retrieval, the gap between simple verdict prediction and evidence-grounded verification remains large. This opens the door to creation of specialized systems that can better navigate the hierarchical and temporal structures of legal evidence.

Limitations

First, CaseFacts relies on expert-synthesized summaries from Oyez rather than full judicial opinions. While this ensures high-quality structured evidence, it abstracts away the complex, long-context reasoning found in raw legal texts, potentially simplifying retrieval compared to a full-text corpus.

Our heuristic approach to identifying overruled cases relies on semantic similarity thresholds. While necessary for computational feasibility, this method inherently misses some cases with low semantic similarity, though we expect the number to be quite small.

While we employ a rigorous pipeline to generate hard negative refuted claims, they remain synthetically derived. They may not fully capture the nuance of human-generated legal misinformation, which often relies on subtle misinterpretation or out-of-context citation rather than the direct negation of legal principles.

Finally, CaseFacts focuses exclusively on U.S. Supreme Court rulings. It does not account for statutory law, regulatory codes, or state-level jurisprudence, which are frequent targets of misinformation in the broader legal domain. This also means that if supreme court rulings are overridden by congressional legislation, we are unable to label them as overruled. Furthermore, this work does not consider legal precedence in other countries.

Ethical Concerns

No Legal Advice and Risk of Harm While CaseFacts aims to improve the reliability of automated fact-checking in the legal domain, the models trained or evaluated on this benchmark are research prototypes and should not be used as a substitute for professional legal counsel. Legal AI systems are prone to hallucinations, and in high-stakes environments, such errors can lead to severe consequences, including financial loss or the deprivation of rights. We emphasize that any downstream application derived from this work should utilize a human-in-the-loop workflow, particularly when verifying claims that influence real-world legal decision-making.

Bias and Harmful Historical Content Our dataset is derived from U.S. Supreme Court rulings via Oyez summaries. Historically, judicial corpora reflect the prejudices of their time. Consequently, the dataset contains assertions regarding slavery, segregation, and disenfranchisement (e.g., claims stating that enslaved persons are property). While these claims are labeled as OVERRULED or contextually specific to valid historical precedents, they represent morally wrong viewpoints which we do not agree with. We retain them because a robust legal fact-checker must be able to identify that such precedents—though once "the law"—are no longer valid. Users should be aware that the dataset contains text describing human rights violations.

Dual Use and Misinformation Our methodology includes a specific pipeline for generating hard negatives. While this is necessary to train robust discriminators, the same techniques could theoretically be misused to generate convincing legal disinformation at scale. However, we believe the benefit of releasing a dataset to detect such misinformation outweighs the risk.

Privacy and Data Usage Our claim generation pipeline explicitly prompts models to remove specific identifiers, such as party names and dates, to focus on general legal principles. This minimizes the risk of generating claims that unintentionally expose sensitive information regarding private individuals involved in historical litigation. All source data is derived from publicly available summaries provided by Oyez.

References

S Ajay Mukund and K. S. Easwarakumar. 2025. Optimizing legal text summarization through dynamic retrieval-augmented generation and domain-specific adaptation. *Symmetry*, 17(5).

Mubashara Akhtar, Oana Cocarascu, and Elena Simperl. 2022. PubHealthTab: A public health table-based dataset for evidence-based fact checking. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1–16, Seattle, United States. Association for Computational Linguistics.

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. FEVEROUS: Fact extraction and VERification over unstructured and structured information. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Avishek Anand, Lijun Lyu, Maximilian Idahl, Yumeng Wang, Jonas Wallat, and Zijian Zhang. 2022. Explainable information retrieval: A survey. *Preprint*, arXiv:2211.02405.

Tobias Braun, Mark Rothermel, Marcus Rohrbach, and Anna Rohrbach. 2024. DEFAME: Dynamic evidence-based fact-checking with multimodal experts. *arXiv preprint arXiv:2412.10510*.

Szu-Ju Chen, Jing Jin, Sheng-Lun Wei, Chien-Hung Chen, and Hsin-Hsi Chen. 2025. Retrieving the right law: Enhancing legal search with style translation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, page 2951–2955, New York, NY, USA. Association for Computing Machinery.

Jacob Devasier, Akshith Reddy Putta, Rishabh Mediratta, and Chengkai Li. 2025. Task-oriented automatic fact-checking with frame-semantics. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 13825–13842, Vienna, Austria. Association for Computational Linguistics.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, and 21 others. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Preprint*, arXiv:2308.11462.

Yinghao Hu, Leilei Gan, Wenyi Xiao, Kun Kuang, and Fei Wu. 2025. Fine-tuning large language models for improving factuality in legal question answering. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4410–4427, Abu Dhabi, UAE. Association for Computational Linguistics.

Larissa Mori, Carlos Sousa de Oliveira, Yuehwern Yih, and Mario Ventresca. 2025. Assessing the performance gap between lexical and semantic models for information retrieval with formulaic legal language. *Preprint*, arXiv:2506.12895.

Mihai Nadăş, Laura Dioşan, and Andreea Tomescu. 2025. Synthetic data generation using large language models: Advances in text and code. *IEEE Access*, 13:134615–134633.

Akshith Reddy Putta, Jacob Devasier, and Chengkai Li. 2025. Claimcheck: Real-time fact-checking with small language models. *Preprint*, arXiv:2510.01226.

Markus Reuter, Tobias Lingenberg, Ruta Liepina, Francesca Lagioia, Marco Lippi, Giovanni Sartor, Andrea Passerini, and Burcu Sayin. 2025. Towards reliable retrieval in RAG systems for large legal datasets. In *Proceedings of the Natural Legal Language Processing Workshop 2025*, pages 17–30, Suzhou, China. Association for Computational Linguistics.

Michael Sejr Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. 2023. AVeriTeC: A dataset for real-world claim verification with evidence from the web. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Preprint*, arXiv:2002.10957.

Yuxia Wang, Minghan Wang, Hasan Iqbal, Georgi N. Georgiev, Jiahui Geng, Iryna Gurevych, and Preslav Nakov. 2025. OpenFactCheck: Building, benchmarking customized fact-checking systems and evaluating the factuality of claims and LLMs. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 11399–11421, Abu Dhabi, UAE. Association for Computational Linguistics.

Koki Wataoka, Tsubasa Takahashi, and Ryokan Ri. 2025. Self-preference bias in llm-as-a-judge. *Preprint*, arXiv:2410.21819.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Jingze Zhang, Jiahe Qian, Yiliang Zhou, and Yifan Peng. 2025a. Enhancing health fact-checking with llm-generated synthetic data. *Preprint*, arXiv:2508.20525.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025b. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *Preprint*, arXiv:2506.05176.

Rongwen Zhao and Jeffrey Flanigan. 2025. SYNTHVERIFY: Enhancing zero-shot claim verification through step-by-step synthetic data generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 3257–3274, Vienna, Austria. Association for Computational Linguistics.

Lucia Zheng, Neel Guha, Brandon R. Anderson, Peter Henderson, and Daniel E. Ho. 2021. When does pretraining help? assessing self-supervised learning for law and the casehold dataset. *Preprint*, arXiv:2104.08671.

A Appendix

A.1 Reproducibility

All of our code is open-sourced at https://github.com/idirlab/supreme-court-dataset. Github Copilot was used in the creation of some of our code.

A.2 LLM Prompts

A.3 Figures

Listing 1: Prompt for Claim Generation

~~~
# Instructions:
Generate truthful, factual claims about the legal implications of this case using simple, everyday
language.
Avoid legal jargon or court-report phrasing. Do not use phrases like "the court found," "the decision
 clarified," or "under this test."
Write claims as clear statements of what the law allows or does not allow.
Adhere to the following rules:
- Do not use any part of the case name or other identifying information in the claim.
- Do not make claims that only apply to the parties in this case; claims must be general legal
principles.
- Keep each claim focused on one central idea and make it easy to understand.
- Each claim must be distinct from the others; do not repeat the same core idea.
- Use direct, plain wording rather than legal formulations.

Output Format:
## Return the claim in a JSON object with the following format:
```json
{{
    "claim1": "...",
    "claim2": "...",
    ...
}}
```

## Facts:
{facts}

# Question:
{question}

# Conclusion:
{conclusion}
~~~

Listing 2: Prompt for the Factuality Check

~~~
You are a legal expert. Read a short claim about a supreme court case plus the case facts, question,
and conclusion. Decide whether the claim is factually consistent with the evidence from the case or
is factually inconsistent with the case evidence.

Rules:
1. Base your judgment only on the Supreme Court evidence.
2. If the evidence does not support the claim, do not label as consistent.
3. Do not rely on outside knowledge or assumptions.
4. Do not invent information that is not in the evidence.

Claim: {claim}

Case Evidence:
Facts: {facts}
Question: {question}
Conclusion: {conclusion}

## Output Format:
Return a JSON object in the following format:
```json
{{
    "explanation": "...",
    "contradiction": "<consistent/inconsistent>",
    ...
}}
```
~~~

Listing 3: Prompt for the Contradiction Check within the same case

~~~
You are a legal expert. Read two short claims about a case plus the case facts, question, and
conclusion. Decide whether the two claims are contradicting (negation or opposite entailment), or
consistent (when they are unrelated or entail).
You must output an explanation for your decision in the "explanation" field. Then, also provide a
decision in the "contradiction" field: "contradiction" if the claims are contradicting, and "
consistent" if they are not.

Claim 1: {claim1}
Claim 2: {claim2}

Case Evidence:
Facts: {facts}
Question: {question}
Conclusion: {conclusion}

## Output Format:
Return a JSON object in the following format:
```json
{{
    "explanation": "...",
    "contradiction": "<contradiction/consistent>",
    ...
}}
```
~~~

Listing 4: Prompt for the Overlap Check within the same case

~~~
You are a legal expert. Read two short claims about a case plus the case facts, question, and
conclusion. Decide whether the two claims are saying the same thing (being redundant) or are
meaningfully different regarding their meaning.
You must output an explanation for your decision in the "explanation" field. Then, also provide a
decision in the "overlap" field: "redundant" if the claims are redundant, and "different" if they are
 not.

Claim 1: {claim1}
Claim 2: {claim2}

Case Evidence:
Facts: {facts}
Question: {question}
Conclusion: {conclusion}

## Output Format:
Return a JSON object in the following format:
```json
{{
    "explanation": "...",
    "overlap": "<redundant/different>",
    ...
}}
~~~

Listing 5: Prompt for Contradiction resolutions within the same case

~~~
You are a legal expert. You are given two claims about the same Supreme Court case that have been
identified as contradictory. Read the two short claims about the case and the case facts, question,
and conclusion.
Your task is to determine which claim is factually correct based on the evidence, or if neither is
correct. In the ”decision” field, only return one of the decision categories.

Claim 1: {claim1}
Claim 2: {claim2}

Reason for contradiction: {explanation}

Case Evidence:
Facts: {facts}
Question: {question}
Conclusion: {conclusion}

Analyze the evidence and decide:
1. Is Claim 1 correct and Claim 2 incorrect?
2. Is Claim 2 correct and Claim 1 incorrect?
3. Are both claims incorrect?
4. Are both claims partially correct but phrased poorly? (If so, provide a merged/corrected claim).

## Output Format:
Return a JSON object in the following format:
```json
{{
    ”explanation”: ”...”,
    ”decision”: ”<claim1_correct | claim2_correct | neither_correct | both_partial>”,
    ”corrected_claim”: ”...” (optional, if decision is ’both_partial’)
}}
~~~
```

Listing 6: Prompt for Overlap resolutions within the same case

```
You are a legal expert. You are given two claims about the same Supreme Court case that are redundant
 (saying the same thing). Read the two short claims about the case and the case facts, question, and
conclusion.
Your task is to choose the one that is better written, more precise, or more comprehensive. You are
also given the reason for redundancy, take that into account when making your decision. In the ”keep”
 json field, return the strings ”claim1” or ”claim2”, not the actual content of the claims.

Claim 1: {claim1}
Claim 2: {claim2}

Reason for redundancy: {explanation}

Case Evidence:
Facts: {facts}
Question: {question}
Conclusion: {conclusion}

Output Format:
Return a JSON object in the following format:
```json
{{
    ”explanation”: ”...”,
    ”keep”: ”<claim1/claim2>”
}}
```
```
```

Listing 7: Prompt for the Contradiction Check within different cases

~~~
You are a legal expert. Read two short claims from different cases and the two cases’ associated
facts, legal questions, and conclusions.
Decide whether the two claims are contradicting (negation or opposite entailment), or consistent (
when they are unrelated or entail).
You must output an explanation for your decision in the "explanation" field. Then, also provide a
decision in the "contradiction" field: "contradiction" if the claims are contradicting, and "
consistent" if they are not.

## Output Format:
Return a JSON object in the following format:
```json
{{
    "explanation": "...",
    "contradiction": "<contradiction/consistent>",
    ...
}}
~~~

Claim 1: {claim1}
Claim 2: {claim2}

Claim 1 Case Evidence:
Facts: {facts1}
Legal Question: {api_question1}
Conclusion: {api_conclusion1}

Claim 2 Case Evidence:
Facts: {facts2}
Legal Question: {api_question2}
Conclusion: {api_conclusion2}
```

Listing 8: Prompt for the Overlap Check within different cases

```
You are a legal expert. Read two short claims from different cases and the two cases’ associated
facts, legal questions, and conclusions.
Decide whether the two claims are saying the same thing (being redundant) or are meaningfully
different regarding their meaning.
You must output an explanation for your decision in the "explanation" field. Then, also provide a
decision in the "overlap" field: "redundant" if the claims are redundant, and "different" if they are
 not.

## Output Format:
Return a JSON object in the following format:
```json
{{
    "explanation": "...",
    "overlap": "<redundant/different>",
    ...
}}
```

Claim 1: {claim1}
Claim 2: {claim2}

Claim 1 Case Evidence:
Facts: {facts1}
Legal Question: {api_question1}
Conclusion: {api_conclusion1}

Claim 2 Case Evidence:
Facts: {facts2}
Legal Question: {api_question2}
Conclusion: {api_conclusion2}
```
```

Listing 9: Prompt for the Contradiction Resolution within different cases

~~~
You are a legal expert. You are given two claims from different Supreme Court cases that have been
identified as contradictory. Read the two short claims about the cases and the two cases’ facts,
question, and conclusion.
1. Are they overruling one another? Indicate ”case1_overruled” if Case 2’s evidence points to
overruling Case 1’s evidence. Indicate ”case2_overruled” if Case 1’s evidence points to overruling
Case 2’s evidence. Take into account their ruling dates when making overruling decisions.
2. Are they consistent given context? (e.g. different jurisdictions, different specific facts).
Indicate ”consistent” in the decision field. Even if the claims are slightly contradicting, they are
consistent as long as they propagate different legal principles. This will be quite common, as true
overruling contradictions are rare in Supreme Court cases.

Claim 1: {claim1}
Claim 2: {claim2}

Case 1 Evidence:
Ruling Date: {date1}
Facts: {facts1}
Question: {api_question1}
Conclusion: {api_conclusion1}


Case 2 Evidence:
Ruling Date: {date2}
Facts: {facts2}
Question: {api_question2}
Conclusion: {api_conclusion2}

Output JSON:
{{
   ”explanation”: ”...”,
   ”decision”: ”case1_overruled” | ”case2_overruled” | ”consistent”,
}}
```
~~~

Listing 10: Prompt for the Overlap Resolution within different cases

```
You are a legal expert. You are given two claims from different cases that have been identified as
redundant (overlapping).
Your task is to decide how to resolve this overlap. You can:
1. Keep Claim 1 (if it is more accurate, comprehensive, or better phrased). The ground truth for
Claim 1 will be both cases.
2. Keep Claim 2 (if it is more accurate, comprehensive, or better phrased). The ground truth for
Claim 2 will be both cases.
3. Merge them (create a new claim that combines the information from both).

Claim 1: {claim1}
Claim 2: {claim2}

Case 1 Evidence:
Facts: {facts1}
Question: {api_question1}
Conclusion: {api_conclusion1}

Case 2 Evidence:
Facts: {facts2}
Question: {api_question2}
Conclusion: {api_conclusion2}

Output JSON:
{{
   ”reasoning”: ”...”,
   ”decision”: ”keep_1” | ”keep_2” | ”merge”,
   ”merged_claim”: ”...” (only if decision is merge, otherwise null)
}}
```

Listing 11: Prompt for the Negations Generation

~~~
You are a legal expert. Read the following claim about a legal case, along with the case’s facts,
question, and conclusion.
Your task is to generate a negation of this claim. The negation should be plausible but factually
incorrect based on the original claim and the case evidence.
Provide an explanation for why this negation contradicts the original claim in the ”explanation”
field. Then, provide the negated claim in the ”negation” field.

Claim: {claim}

Case Evidence:
Facts: {facts}
Question: {question}
Conclusion: {conclusion}

## Output Format:
Return a JSON object in the following format:
```json
{{
    ”explanation”: ”...”,
    ”negation”: ”...”
}}
```
~~~

Listing 12: Prompt for the Negations Length Fixing

~~~
# Instructions:
Rewrite the following legal claim to be a concise, general legal principle.
The input claim is a ”refuted” (false) legal claim, but it is currently might be too long, specific,
or conditional. The case the claim originated from is provided as context. It will contradict the
claim, do not change the meaning of the claim.
Your task is to rewrite it so it is:
1. Independent of specific case details or parties (remove names, dates, specific locations).
2. Unconditional (remove ”unless”, ”especially if”, or specific factual caveats).
3. Concise and direct (simple, everyday language).
4. Focused on the core legal principle being asserted (even if that principle is false).

You must not change the meaning of the claim. If some details are necessary to preserve the meaning,
keep them, even if that makes the claim lengthy.

If the claim is already concise and general, you may return it as is or with minor improvements.

## Input Claim:
”{claim}”

## Output Format:
Return a JSON object with a single key ”rewritten_claim”:
```json
{{
    ”rewritten_claim”: ”...”
}}
~~~

Listing 13: Prompt for the Naive Factuality check

~~~
You are a legal expert. Read a short claim about a supreme court case. Decide whether the claim is
factually consistent with the details of the case.

Rules:
1. Base your judgment on your internal knowledge of the Supreme Court case.
2. If you are unsure or do not know the case, do not label as consistent.

Claim: {claim}

## Output Format:
Return a JSON object in the following format:
```json
{{
    ”explanation”: ”...”,
    ”contradiction”: ”<consistent/inconsistent>”,
    ...
}}
```
~~~

Listing 14: Prompt for Baseline Predictions

```
You are a legal expert. Your task is to analyze a legal claim and determine its veracity based on US
Supreme Court cases.
You must determine if the claim is ”Supported”, ”Refuted”, or ”Overruled” by the case law.
You must also identify the specific Supreme Court cases that serve as evidence for your decision.
List them in order of importance (most important first). You will be penalized for irrelevant and
incorrect citations, so prioritize accuracy and conciseness of citations.

Constraints:
1. You must ONLY cite cases from the provided list of valid Supreme Court cases. Do not invent cases
or cite cases not in the list.
2. Do not guess. If you are unsure, provide your best estimate but prioritize accuracy.
3. Output must be a valid JSON object.

Valid Supreme Court Cases:
{case_list}

Claim: {claim}

Respond with a JSON object in the following format:
{{
    ”explanation”: ”Brief explanation of your reasoning.”,
    ”cases”: [”Case Name 1”, ”Case Name 2”, ...],
    ”verdict”: ”Supported” or ”Refuted” or ”Overruled”
}}
```

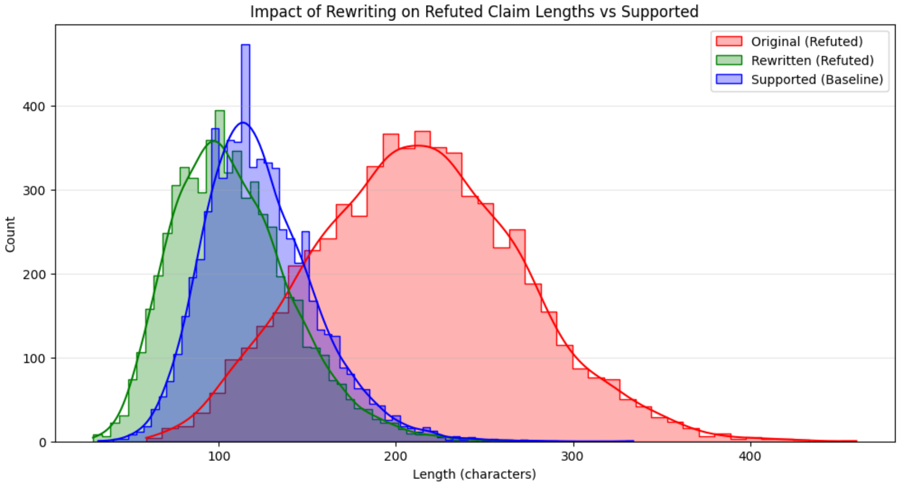

Figure 1: Graph displaying the changes in character length before and after the LLM pass with prompt 12. The length of the supported claims, which negations are generated from, is provided for comparison.