Operationalizing Evidentiary Risk in Multi-Turn AI Conversations: A Case Study Based on Tennessee SB1493 / HB1455

Emmanuel Donate

February 14, 2026

Abstract

Recent legislative efforts regulating artificial intelligence systems increasingly rely on statutory language that references conversational behaviors, user engagement patterns, and relational characteristics of AI interactions. This paper presents a concrete, reproducible evaluation method for generating litigation-style evidentiary artifacts from realistic multi-turn AI conversations. Using Tennessee Senate Bill 1493 / House Bill 1455 as a case study, the paper demonstrates how structured conversation replay, feature extraction, rule-based flagging, and independent model-based assessment can be combined to produce artifacts suitable for legal, regulatory, or risk analysis review. The contribution is methodological rather than normative: the paper does not argue for or against the statute, nor does it assess legal compliance. Instead, it shows how conversational logs could plausibly be transformed into evidence-like records under statutes written in a conversationally grounded manner.

1 Introduction

AI evaluation has historically focused on model capabilities, safety failures, or benchmark performance. However, recent legislative language has begun to reference interaction-level characteristics of AI systems, such as emotional acknowledgement, invitations to continue engagement, or step-by-step guidance, creating new evaluation challenges that are not addressed by standard benchmarks. These properties are not easily captured by traditional benchmarks, yet they may become relevant in legal or regulatory contexts.

This paper addresses a narrow but increasingly important question: how could realistic multi-turn chatbot conversations be transformed into evidentiary artifacts that could plausibly be presented, scrutinized, and contested in a legal proceeding under statutes that regulate conversational behavior rather than model internals?

Rather than arguing for a particular legal interpretation or regulatory outcome, this paper presents a concrete method for turning realistic chatbot conversations into records that could plausibly be examined as evidence.

2 Statutory Context

Tennessee Senate Bill 1493 / House Bill 1455 was submitted in December 2023 by State Senator Becky Massey (R-Knoxville) and State Representative Mary Littleton (R-Dickson) [1].

The statute addresses the regulation of certain AI-driven conversational systems and is notable for referencing behavioral and interactional characteristics rather than purely technical system attributes. While the legal interpretation of such language is outside the scope of this paper, its structure motivates the need for evaluation approaches that operate at the level of conversational transcripts.

The statute is treated here solely as a motivating example of a broader class of regulations that may rely on conversational evidence.

3 Problem Framing

A central challenge posed by conversationally grounded statutes is that potential evidence does not reside in a single model output, but across multi-turn interactions that evolve over time.

Key questions include:

- How can conversational behavior be replayed deterministically?
- How can specific interaction patterns be identified without relying on opaque scoring?
- How can outputs be logged in a form suitable for external review?
- How can rule-based and model-based assessments coexist without conflation?

The goal is not to determine legality, but to create artifacts that could plausibly be examined by third parties such as attorneys, regulators, or auditors.

4 Method Overview

The evaluation harness presented here consists of five stages:

1. Deterministic conversation replay using a structured conversation tree
2. Full transcript logging with turn indices and metadata
3. Interpretable feature extraction with quoted excerpts
4. Rule-based evidentiary flagging
5. Independent judge-model assessment

Each stage produces a standalone artifact, enabling inspection without reliance on internal system state.

A deliberate design choice is the use of two independent assessment mechanisms. The rule-based flag (stage 4) is fully deterministic and transparent: it applies a fixed boolean formula over pattern-detected features, producing a result that is auditable, reproducible, and explainable without reference to any model. The judge-model assessment (stage 5) provides a complementary evaluative lens that can detect patterns the substring rules miss, but whose outputs are inherently non-deterministic and harder to audit. By logging both assessments as separate artifacts with full provenance, the system avoids conflating mechanistic detection with model-mediated judgment. This separation is motivated by the evidentiary context: a reviewer examining these artifacts can independently evaluate the strength of each assessment and identify cases where they diverge, which may be more informative than either assessment alone.

5 Conversation Scenario

The implemented scenario involves a landlord-tenant dispute in which a user seeks guidance regarding a withheld security deposit. The conversation unfolds across five phases:

1. Orientation
2. Clarification
3. Procedural follow-up
4. Relational cue
5. Continuation request

Branching is deterministic and controlled via an environment variable, ensuring reproducibility while preserving realistic conversational flow. At the continuation phase, the scenario supports three branch conditions (`ask_for_guidance`, `ask_for_support`, `ask_for_plan`), enabling comparison across different conversational paths against the same target model.

Importantly, assistant nodes in the conversation tree specify *content intent* rather than literal text. The target LLM generates the actual response at runtime, which is what gets evaluated. This ensures the harness tests the model's natural conversational behavior rather than canned responses.

6 Feature Extraction

Rather than counting occurrences or assigning numeric scores, the system performs presence-based pattern detection. Features are defined as observable conversational behaviors, including:

- Acknowledgement of user emotion
- Invitations to continue or disclose more
- Procedural or step-by-step guidance
- Relational framing
- Offers of continued engagement

Each detected feature is accompanied by:

- The turn index
- A short quoted excerpt (90-character window around the matched phrase)
- A feature label

Relational and continuation-style features are *phase-gated*: they only fire in later conversation phases (relational or continuation) where the conversational context makes relational behavior meaningful. This prevents false positives from early-phase matches where phrases like “let me know” or “I can help” are procedural rather than relational.

This design prioritizes interpretability over optimization. Substring matching is intentionally chosen over ML-based classifiers because the features must be fully auditable in a legal context.

7 Rule-Based Evidentiary Flag

A simple rule-based layer aggregates feature presence into a boolean evidentiary flag. The flag requires a conjunction: acknowledgement of emotion **and** invitation to continue **and** structured guidance **and** either relational framing or a continued engagement offer must all be present.

The flag indicates whether a transcript contains a combination of behaviors that could plausibly be offered as evidence of emotionally engaged, ongoing interaction. Importantly, the rule does not claim statutory violation or compliance. It merely identifies transcripts that may warrant closer examination.

8 Judge Model Assessment

In parallel, an independent language model is tasked with reading the full transcript and returning a structured JSON assessment including:

- A categorical score (`likely_yes`, `borderline`, `likely_no`)
- A short rationale (2–3 sentences)
- Cited turn indices

The judge model is treated as an evaluative lens, not a ground truth authority. Its output is logged verbatim and validated for structural correctness. If the judge returns malformed JSON, the system falls back to a `borderline` score with `valid_json: false` rather than crashing, preserving the pipeline while flagging the degradation.

This approach draws on the emerging practice of using language models as evaluators of other language model outputs [2], while maintaining the principle that no single assessment mechanism is authoritative.

9 Implementation

The evaluation pipeline is implemented in Python and built on the `inspect-ai` framework [3], which provides model abstraction, asynchronous generation, and solver composition.

The pipeline is strictly linear, orchestrated by a single runner module:

1. A YAML conversation tree is loaded and parsed. The runner navigates the tree node by node, injecting scripted user messages and calling the target model for assistant turns.
2. Each turn (system, user, assistant) is appended to a JSONL transcript file with turn index, role, content, node ID, and conversation phase.
3. After replay completes, the transcript is read back and passed through the feature extraction layer, which scans only assistant turns.
4. The extracted features are passed to the rule-based flag, which applies its boolean formula.
5. The full transcript is serialized as numbered text and sent to the judge model for independent assessment.
6. A Markdown evidentiary memo is assembled, merging rule-based evidence and judge-cited turns into a single excerpt table with counterarguments.

All artifacts are written to a timestamped run directory (`outputs/YYYYMMDD_HHMMSS/`), ensuring each run is self-contained and reproducible. Configuration is controlled entirely through environment variables, and the system validates required API keys at startup.

10 Generated Artifacts

Each evaluation run produces the following artifacts:

- `transcript.jsonl`: full turn-by-turn conversation log
- `run_meta.json`: run configuration and metadata
- `features.json`: extracted features with excerpts
- `a6_rule.json`: rule-based evidentiary flag
- `a6_judge.json`: judge model assessment
- `memo.md`: neutral evidentiary memorandum

All artifacts are stored in a run-specific directory to preserve auditability.

11 Evidentiary Memo

The generated memorandum adopts a litigation-style format, summarizing:

- Run metadata
- Rule-based findings
- Judge assessment
- Quoted excerpts with turn numbers
- Counterarguments and limitations

The memo is intentionally neutral in tone and avoids legal conclusions.

12 Limitations

This case study has several limitations:

- Only a single statute and scenario are evaluated
- Feature definitions are intentionally conservative and rely on substring matching
- No claim is made about enforcement likelihood or legal sufficiency
- The judge model assessment is non-deterministic and subject to the limitations of LLM-as-judge approaches [2]
- The approach does not generalize automatically to all conversational statutes

These limitations are by design, prioritizing clarity and reproducibility over breadth. The feature extraction patterns, rule logic, and report counterarguments are currently hardcoded to the A6 landlord-tenant scenario; generalizing to additional statutes or scenarios would require refactoring these components.

13 Discussion

The primary contribution of this work is not a legal argument or a policy recommendation, but a demonstration that conversational evaluation can be operationalized in a manner compatible with evidentiary review.

For AI practitioners, the work highlights a class of risks that are not captured by standard benchmarks. For evaluators, it illustrates how structured logging and interpretation-first design can support downstream scrutiny. For legal researchers, the dual-assessment approach (deterministic rule plus independent model judge) provides a template for producing artifacts where the provenance and limitations of each finding are explicit.

14 Conclusion

As AI regulation increasingly engages with conversational behavior, evaluation methods must extend beyond static prompts and aggregate scores. This paper presents a concrete, inspectable approach for generating evidentiary artifacts from realistic multi-turn interactions.

While the case study is grounded in a specific Tennessee statute, the method is applicable to a broader class of conversationally framed regulatory regimes. Future work may explore additional statutes, scenarios, and evaluation lenses, but the core contribution remains methodological: showing how conversational evidence can be produced, logged, and examined with rigor.

Non-Claims

This paper does not provide legal advice, does not interpret statutory intent, and does not assess compliance or violation. All artifacts are generated for evaluation and research purposes only.

References

[1] Tennessee Senate Bill 1493 / House Bill 1455. `https://www.capitol.tn.gov/Bills/114/Bill/SB1493.pdf`

[2] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *Advances in Neural Information Processing Systems*, 36, 2023.

[3] UK AI Safety Institute. Inspect: A framework for large language model evaluations. `https://inspect.ai-safety-institute.org.uk/`, 2024.