

Graph-Based Validation and Review of AI-Generated Document Analysis

Technical input concerning proposed Rule 707 and AI-related validation under Rule 702

Moriya Dechtiar | 23 September 2026

Submitted in my personal capacity; the views expressed are my own.

Proposal to the Committee

I recommend guidance on how document structure is captured, represented, and checked when validating AI-generated conclusions. My published methods for constructing and evaluating contract graphs provide components for this procedure. The proposed record connects source provisions with their relationships and reports unresolved dependencies and measured checker performance.

Document structure carries information that affects how its text is interpreted and what conclusions follow. Clause hierarchy, definition scope, cross-references, and dependencies connect provisions whose meaning can change when read together. AI systems should capture and represent this structural information alongside the text when generating and validating document-based conclusions. A correct quotation alone can miss a condition elsewhere. [5]

My coauthored research contributes graph representation and structural inspection, evaluation and uncertainty-based triage of contract graphs, and methods for defining and testing contract-smell categories. These components support the proposed workflow. [5–7]

The draft May 7, 2026 minutes record the Committee’s interest in how litigants may demonstrate the reliability of AI-generated conclusions. The May 17 report’s revised Note identifies “analytical error or incompleteness,” relevant testing conditions, and auditability as concerns. [1]

A practical validation workflow

1. Map the document. Connect clauses and their hierarchy with definitions, parties, values, and references. Preserve a source location for each extracted relationship and check extraction fidelity. [5]

2. Run a contract linter. Use automated checks to flag potential drafting and structural issues: missing references, undefined or unused terms, and suspicious dependency patterns. Inspect flagged passages against the source. [5, 7]

3. Trace the conclusion. Examine the cited provision together with linked definitions, qualifications, and incoming as well as outgoing references. Record the source provisions reviewed, the checks performed, and any unresolved dependencies.

4. Measure review reliability. Report errors detected, errors missed, correct outputs unnecessarily flagged, and errors remaining after review or correction. Evaluate under conditions relevant to the proposed use. [6]

Suggested Committee Note language. For an AI-generated conclusion drawn from documents, an accurate quotation does not by itself establish that the conclusion accounts for material context. Relevant considerations include how document structure and material dependencies, such as definitions, qualifications, cross-references, and amendments, were represented and checked. Validation evidence should distinguish the accuracy of extracted information and resulting conclusions from the performance of any checking or review process, including errors left undetected under conditions comparable to the proposed use. An inspectable representation of document relationships, such as a graph, can assist this assessment.

A correct quotation can accompany a wrong conclusion

Constructed illustration. Invented provisions, facts, and an answer demonstrate the dependency-checking problem.

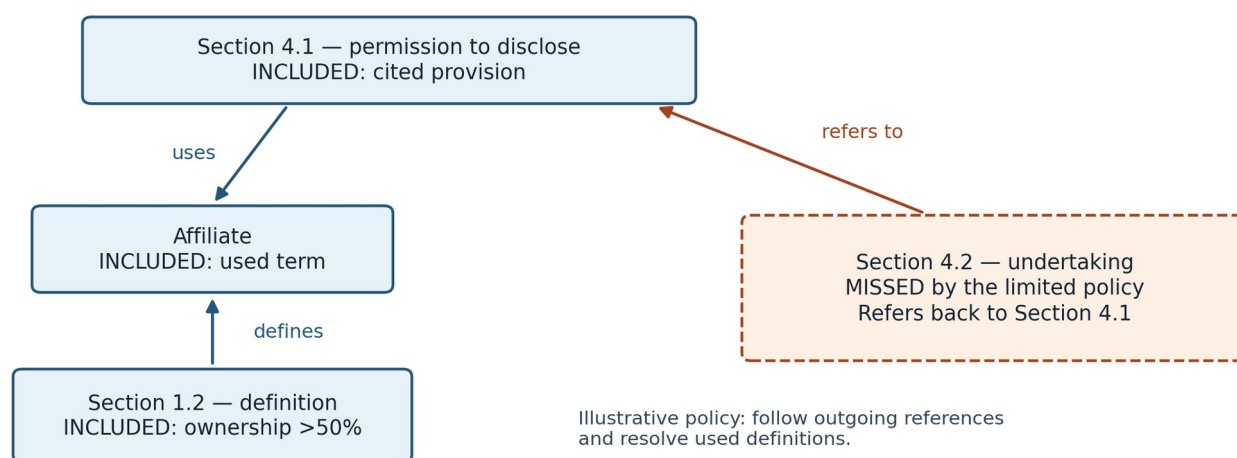
§1.2 “Affiliate” means an entity in which Recipient owns more than fifty percent of the voting interests.

§4.1 Recipient may disclose Confidential Information to an Affiliate.

§4.2 The permission in Section 4.1 applies only if the Affiliate is bound by a written confidentiality undertaking.

Question Recipient owns 60% of Company B’s voting interests. B is not bound by a written confidentiality undertaking. Under these stipulated terms and facts, may Recipient disclose?

Illustrative faulty answer Yes. Section 4.1 states, “Recipient may disclose Confidential Information to an Affiliate.”



Check	Outcome in this illustration
Source location	Pass: the cited provision exists.
Quotation accuracy	Pass: its wording is reproduced accurately.
Excerpt-only apparent support	The excerpt states a permission; §4.2 supplies its condition.
Contextual sufficiency	Fail: the answer omits an unsatisfied condition in §4.2.

Under these stipulated terms and facts, disclosure is not permitted: the ownership threshold is satisfied, but the written-undertaking condition is not. Following only outgoing references from §4.1 and resolving its used definition reaches §1.2 while missing §4.2’s incoming reference. Inspecting that incoming connection exposes the missing condition. The figure shows document relationships and a limited retrieval policy.

What the research contributes to document analysis

Three published building blocks

Represent and inspect the document's structure. GRAPH-GRPO-LEX develops an ontology and extraction pipeline linking clauses, hierarchy, terms, parties, values, and references. Its contract linter examines the resulting graph through definition-coverage checks, dependency tracing, cycle detection, and connectivity and articulation analysis. The Zogenix case study reports 257 nodes, 916 edges, 131 orphan nodes (zero incoming edges), 11 articulation points, and a dependency depth of six. These diagnostics identify structural features to inspect for possible missing links and critical dependencies. [5]

Descriptive graph statistics from one agreement; not counts of confirmed drafting defects or error rates on legal conclusions.

Measure which outputs need attention

My coauthored LLM-as-a-Judge study combines structural, semantic, and LLM-based evaluators, measures their disagreement, and calibrates error probabilities to prioritize unreliable contract graphs. In Table 3, selecting 15 of 150 outputs for review captured 91% of labeled errors, with 67% precision. The principal evaluation uses reference graphs. The study also evaluates outputs after refinement, providing a basis for measuring both review selection and the effect of correction. [6]

Define and evaluate review categories. Software engineering meets legal texts defines contract-smell categories, creates clause datasets using automated labeling with expert spot-checks, and evaluates detection performance by category. It turns specified concerns into testable review tasks; the spot-checks assess label consistency with the category definitions. [7]

Relevance beyond contracts. A technical report may qualify a finding in an appendix; a medical record may contain a later correction; and a document may incorporate an attached exhibit. These examples illustrate how relationships within and between documents can affect an AI-generated conclusion. The empirical evidence here comes from contracts.

Scope of the evidence. The experiments concern contract-smell detection and contract-graph extraction and evaluation. Whole-answer accuracy, integration of the proposed workflow, and transfer to other documents and tasks require further evaluation. Graph alerts identify matters for source review; legal applicability and admissibility remain for qualified reviewers and the court.

A real example and the wider research context

ContractNLI development document 610 is the Nimble Storage–Hewlett Packard Enterprise NDA. Section 1 limits use for a proposed transaction while preserving permissions elsewhere; §8 permits use of residual knowledge subject to restrictions. For the dataset's limited-use hypothesis, the supplied label is Contradiction and the annotated evidence includes both provisions, spans 19 and 46. This illustrates why review must account for separated provisions. Appendix A gives the exact record. [3–4]

The proposal builds on ContractNLI's document-level inference, ALCE's separation of citation quality from answer correctness, LeMAJ's omission detection and review triage, and Cormack and Grossman's evaluation of recall and review effort. It contributes a structural account of document dependencies to the broader validation and disclosure discussion, including the recent Lawyers for Civil Justice submission. [3, 8–11]

Requested action. Incorporate a contextual-completeness factor in the Committee Note, along the lines proposed above, with an inspectable dependency record and separate measures of extraction, conclusion, and review performance.

References and source locations

- [1] Advisory Committee on Evidence Rules. Report to the Standing Committee, May 17, 2026, printed pp. 7–11: revised proposed Rule 707 and Note. See also the attached draft minutes of May 7, 2026, p. 16 (PDF p. 31); the report identifies the minutes as not yet approved (report p. 15).
https://www.uscourts.gov/sites/default/files/document/advisory_committee_on_evidence_rules_may_2026.pdf
- [2] Federal Rules of Evidence, Rule 702 (current rules pamphlet, printed pp. 15–16).
<https://www.uscourts.gov/sites/default/files/document/federal-rules-of-evidence.pdf>
- [3] Yuta Koreeda and Christopher D. Manning. ContractNLI: A Dataset for Document-level Natural Language Inference for Contracts. Findings of EMNLP 2021, pp. 1907–1919, especially §5.2. Official release: development document 610, hypothesis nda-4.
<https://aclanthology.org/2021.findings-emnlp.164/>
- [4] Nimble Storage, Inc. and Hewlett Packard Enterprise Company. Mutual Non-Disclosure Agreement, November 17, 2016, §§1, 8; SEC Exhibit (e)(3).
<https://www.sec.gov/Archives/edgar/data/1452751/000119312517086167/d334994dex99e3.htm>
- [5] Moriya Dechtiar, Daniel Martin Katz, Mari Sundaresan, Sylvain Jaume, and Hongming Wang. GRAPH-GRPO-LEX: Contract Graph Modeling and Reinforcement Learning with Group Relative Policy Optimization. IEEE ICDM Workshops 2025, pp. 767–776; §§IV.A.3 and V.B (author manuscript, pp. 3–4 and 6–7). DOI: 10.1109/ICDMW69685.2025.00092.
<https://arxiv.org/abs/2511.06618>
- [6] Moriya Dechtiar, Daniel Martin Katz, Sylvain Jaume, and Hongming Wang. LLM as a Judge for Evaluating Contract Graphs: Multi-Judge Benchmarking and Agentic Uncertainty-Aware Refinement. *Procedia Computer Science* 275 (2026), pp. 513–522 (ACLing 2025). Table 3; reported figures checked against p. 7 of the author manuscript.
<https://doi.org/10.1016/j.procs.2026.01.060>
- [7] Moriya Dechtiar, Daniel Martin Katz, and Hongming Wang. Software engineering meets legal texts: LLMs for auto detection of contract smells. *Machine Learning with Applications* (2025), article 100639.
<https://doi.org/10.1016/j.mlwa.2025.100639>
- [8] Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling Large Language Models to Generate Text with Citations. EMNLP 2023, pp. 6465–6488, §§3.2–3.3.
<https://aclanthology.org/2023.emnlp-main.398/>
- [9] Joseph Enguehard et al. LeMAJ (Legal LLM-as-a-Judge): Bridging Legal Reasoning and LLM Evaluation. NLLP 2025, pp. 318–337, §§3.1–3.2 and 5.
<https://aclanthology.org/2025.nllp-1.23/>
- [10] Gordon V. Cormack and Maura R. Grossman. Autonomy and Reliability of Continuous Active Learning for Technology-Assisted Review (2015).
<https://arxiv.org/abs/1504.06868>
- [11] Lawyers for Civil Justice. September 3, 2026 submission on Rules 707 and 901(c), docket 26-EV-6, especially §§I.A, I.C, I.D, and I.F.
https://www.uscourts.gov/sites/default/files/document/26-ev-6_suggestion_from_lcj_-_rule_707_and_901c.pdf

Prepared by Moriya Dechtiar in her personal capacity. Citation of coauthored research does not imply coauthor participation in or endorsement of this submission.

Appendix A. Document-dependency record

The record connects source text, document structure, and semantic entities so a reviewer can inspect a conclusion's relevant context. Each relationship should retain its source basis and identify any inference involved. The fields below support implementation in a graph or another representation that preserves these connections.

Record	Information needed to inspect it
Source	Document identifier and version; available exhibits and amendments; stable copy or content hash where useful.
Text	Original provision identifiers, exact spans or page locations, parent provision, and defined-term scope.
Relationship	Linked provisions or entities, relation type, supporting spans, extraction method and version.
Interpretation	Whether the relation is explicit, structurally inferred, or semantically inferred; unresolved ambiguity and validation status.
Missing context	Unresolved references, absent schedules, conflicting definitions, unavailable amendments, and limits of the search.
Claim and facts	Exact answer and citations; supplied scenario facts kept separate from contractual text.
Actual inputs	Material supplied to the generator and checker; direction and depth of reference search; truncation and other resource limits.
Reconstruction	Whether the record existed at execution time or was reconstructed later; actual model inputs recorded separately.

In the constructed example, the published extraction types USES and DEFINES connect §4.1 and §1.2 through Affiliate; resolving the defining clause requires following DEFINES in reverse. REFERENCES connects §4.2 to §4.1. A proposed semantic label such as QUALIFIES would add an interpretation of that reference and should be separately supported and validated against the source. [5]

Reproducing the real source example

Dataset hypothesis nda-4: "Receiving Party shall not use any Confidential Information for any purpose other than the purposes stated in Agreement." [3]

ContractNLI dev.json, document 610, hypothesis nda-4, supplied label Contradiction; annotated evidence spans 19 and 46. Character offsets are zero-based, end-exclusive: [4075, 4326) and [11051, 11297). These offsets identify text in the released dataset, not coordinates on the SEC webpage.

Spans 47–51 supply further restrictions on residual knowledge, including exclusions and narrow construction. They are additional source context; spans 19 and 46 are the dataset's evidence annotations for nda-4. The dataset authors' annotation process supplied these labels. [3–4]

Dataset repository revision: eced6528dd3c1d14d73f9a87df8f7bdbc03126f9

Official dataset and license: [ContractNLI release \(CC BY 4.0\)](#)

Pinned archive: [contract-nli.zip](#)

Appendix B. Measuring the review process

Measure answer correctness and checker performance separately against a stated reference standard. Identify whether the checker had access to reference graphs or labels, and distinguish benchmark conditions from the proposed operational setting.

A worked reporting example

Hypothetical reporting example; not an empirical result. Of 1,000 answers, a reference assessment identifies 100 as incorrect. The checker flags 150 answers for review.

Reference outcome	Flagged	Unflagged	Total
Incorrect	90	10	100
Correct	60	840	900
Total	150	850	1,000

Error recall: $90 / 100 = 90\%$ of incorrect answers are flagged.

Flag precision: $90 / 150 = 60\%$ of flagged answers are incorrect.

Review allocation: $150 / 1,000 = 15\%$ of answers selected. Record actual review time separately if measured.

Residual error among unflagged answers: $10 / 850 = 1.18\%$.

False-flag rate: $60 / 900 = 6.67\%$ of correct answers are unnecessarily flagged.

Report the underlying counts and uncertainty. The 1.18% is the error fraction in the unflagged set, not the measured error rate of final outputs after review or correction. Score those final outputs separately, including any errors introduced during correction. Confidence in an individual answer is a separate question.

Make the test conditions reproducible

State the error definition, reference-label provenance, sample selection, contract-level splits, model and prompt versions, decision thresholds, source material, and limits on context or calls. Account for observations sharing a contract when estimating uncertainty. Describe the population represented by the sample.

Evaluate each stage: fidelity of the extracted document graph; identification of relevant dependencies; use of those dependencies in the answer; and selection and correction of erroneous outputs. Record unresolved references and source-review findings alongside quantitative results. Establish accuracy against reference assessments independently of model agreement.

Test the proposed workflow against alternatives

A comparative study should include a full-document evaluator, reference expansion, and a nongraph comparator receiving the same passages. Match or report context and model-call budgets, including representation overhead. A second generator and checker family would test transfer beyond one pairing. These comparisons would isolate the contribution of graph organization to answer verification.

The resulting report should let a reader reconstruct what was checked, inspect the supporting document relationships, and assess the errors remaining after review.