Human-in-the-Loop Nugget Annotation for Accountable LLM-as-a-Judge Evaluations

Laura Dietz
University of New Hampshire
USA
dietz@cs.unh.edu

Abstract

Evaluating AI/Agentic system outputs reliably requires human judgment, but how one incorporates the human determines whether one gets a real quality signal or expensive theater. The common approaches either accidentally anchor human experts (leading to rubber-stamping) or leave them unsupported in cognitively demanding labeling tasks. We present a prototype of an annotation tool that implements a different division of labor: humans identify what information matters (nuggets), while LLMs handle high-volume matching of nuggets to system outputs. This plays to each party's strengths while maintaining genuine human oversight. We describe the Human-AI workflow, key design decisions, and how resulting nugget banks are used with automated judges.[1]

Keywords

LLM-as-judge, RAG evaluation, human-in-the-loop

1 Introduction

Using an LLM to judge another LLM's output leads to circular effects that render the evaluation invalid [2]. When the judge shares biases, training data, or architectural patterns with the system being evaluated, the resulting scores conflate "sounds like what an LLM would write" with "actually addresses the user's information need well."

This failure mode is not related to the quality of the underlying LLM. It is merely a result from using the same approach to obtain results and to evaluate—akin to a fifth-grader grading their own essays. The effect has been empirically confirmed [2, 6].

Two common ways to respond to this circularity both fail in different ways:

Approach 1: Human verification of AI proposals. The LLM proposes a decision, then a human reviews and corrects if necessary. This triggers anchoring bias [28] in humans, negatively affecting the accountability of the human-made decision. The problem is that the human sees the machine's answer before forming their own opinion. Research shows this pattern leads to blind agreement even when the judge is wrong [1, 12]. This problem is described as the Rubber-Stamp Effect (Judge Trope #12, [8]): humans under time pressure tend to believe the AI rather than providing critical oversight. A second problem is triage: if the system routes only low-confidence cases to experts, uncalibrated confidence can send humans the wrong data points while leaving errors made with high confidence unchecked [13, 25]. Such contaminated judgments create the conditions for semantic drift: iterative learning can amplify errors in low-quality labels, and evaluations of semi-supervised learning show that adding weak or mismatched data can degrade rather than improve performance [3, 15, 20].

Approach 2: Manual test sets. Humans provide relevance assessments on examples without seeing AI scores, with the goal of replacing, calibrating or evaluating the LLM judge. This approach is a safe way to remove anchoring but leaves the human expert completely unsupported. Assigning a single numerical quality score to a lengthy response for a complex task is a cognitively demanding, high-effort task. This is referred to as Black-box Labeling (Judge Trope #13 [7]): when the criterion is complex, the label becomes hard to produce and hard to interpret. Shankar et al. [26] document criteria drift, where annotators refine their standards as they work, making the benchmark unreliable. It is well-known that even highly trained human experts suffer from fatigue, can provide inconsistent assessments, and sometimes are missing important details. As a result, fully manual benchmarks tend to contain many data annotation errors that negatively affect the evaluation results.

Approach 3: Humans specify what is relevant, AI matches at scale. In this paper we adopt a third approach: human experts decide what information matters, and the AI is restricted to detecting when system outputs match the nugget. This preserves the central accountability of human judgment while using the LLM for the narrower linguistic task it performs well: matching semantically equivalent statements across many responses. The key design principle is therefore not to ask the LLM to decide what is relevant, but to let it operationalize criteria that humans have already articulated.

We discuss LLM judges in the context of retrieval-augmented generation systems, where an information system responds to a query with a long-form answer intended to contain relevant content. The approaches we discuss translate directly to quality measurement for many other AI applications, including multi-turn conversational agents, LLM/harness traces, image generation, and code generation. In each case, some user input leads an AI system to produce an output that is to be evaluated for correctness. By tracking the pieces of information relevant to that particular input or query, we obtain an evaluation measure that represents response quality.

2 Nugget-based LLM Judges

Our approach builds on prior work on nugget-based LLM Judges. A nugget judge evaluates a system response by checking whether it contains a set of predefined query-specific atomic information needs, rather than asking for one holistic relevance or quality score. These pieces of information are commonly referred to as "nuggets" or "SCUs", and describe facts, constraints, or errors that an ideal

[1] Web demo: https://trec-auto-judge.cs.unh.edu/annotate/nugget-hil-demo.html

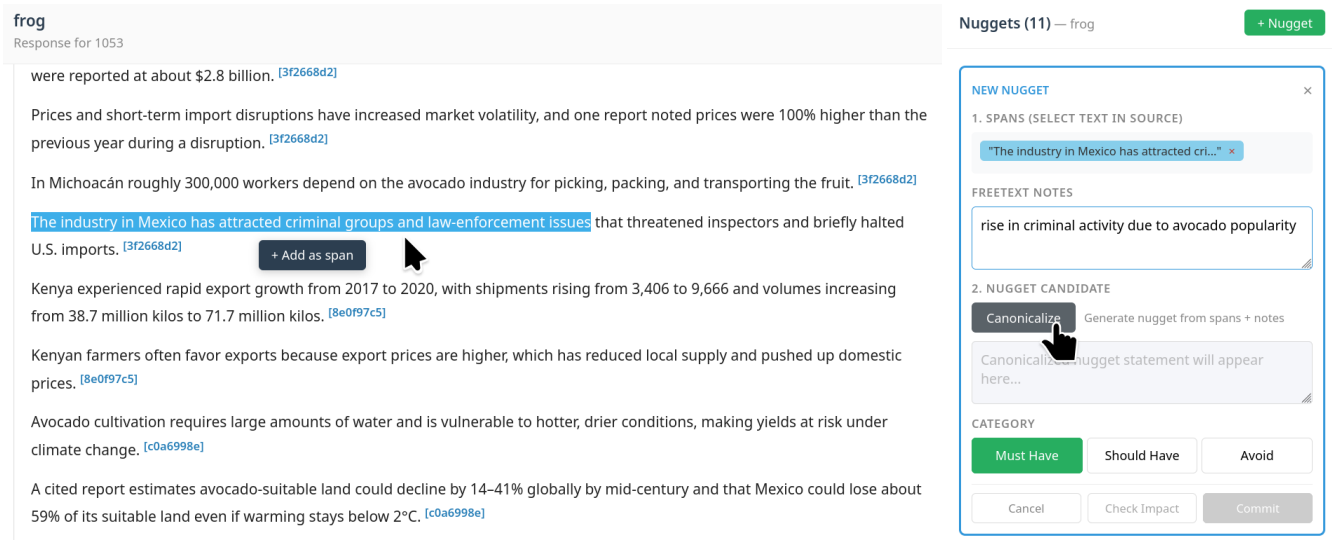

Figure 1: Grounding and note-taking for manual nugget curation. The human expert selects "The industry in Mexico has attracted criminal groups and law-enforcement issues" and types the free-text note: "rise in criminal activity due to avocado popularity." These notes describe their intent of an important piece of information before any AI involvement.

response for a given query should contain or avoid. Unlike holistic relevance scores, nuggets are:

- **Explicit:** The evaluator articulates what matters, rather than relying on an overall impression of response quality.
- **Verifiable:** Each nugget can be checked independently against a response.
- **Reusable:** For a fixed set of test queries, the same nugget banks can be applied consistently across all systems.
- **Auditable:** Disagreements can be traced to specific information pieces.
- **Specific:** Each nugget captures one essential piece of information for a given query. Nuggets do not transfer across queries, but the resulting measure of system performance does generalize.

A collection of nuggets for a query forms a nugget bank. Each nugget in the bank is assigned an importance category that determines how it contributes to the final evaluation:

Must Have: Critical information. Responses that lack this information are not acceptable.
Should Have: Important and nice-to-have information.
Avoid: Information that is wrong, off-topic, or otherwise undesirable—an anti-nugget.

The role of the LLM judge is then restricted to matching: for every system response, the judge determines whether the response expresses each nugget, and records the matching grade and supporting evidence. Must-have and should-have nuggets reward responses that cover the desired information, while avoid nuggets penalize responses that include undesirable information. This turns a complex evaluation task into several smaller, auditable matching decisions.

Once all system responses are graded against all nuggets, the nugget bank can be used to compute commonly used nugget-based evaluation metrics: The average nugget grade summarizes how strongly a response satisfies the all nuggets. Nugget coverage measures how many must-have and should-have nuggets are addressed. Weighted scores combine nugget grades with the must/should/avoid importance categories, allowing critical omissions or harmful inclusions to matter more than missing optional details. These metrics are established ways to compare system quality in nugget-based evaluation [11, 29, 30].

The advantage of this process is that human experts are intellectually in charge of making decisions about what must be included in an ideal system response. At the same time, AI is providing scale and consistency—and even repeatability. Hence, both humans and AI are playing to their respective strengths.

3 Our Approach: Supporting Human Opinion Formation

Given this division of labor, the remaining design question is how to support humans while they identify nuggets. In our design, accountability means that every evaluation criterion can be traced to a human action: a selected span, a free-text note, a category assignment, or an explicit edit. The interface should help experts form their own judgments first, then use AI only for narrowly scoped assistance where humans are prone to inconsistency, omission, or fatigue. The LLM may formalize or preview the consequences of a human-authored nugget, but it cannot introduce evaluation criteria on its own.

3.1 Supporting Accountable Nugget Curation

Humans recognize a good response when they see it, but find it difficult to give an all-encompassing list of must-haves without grounding. To help the human in recognizing relevant information, concrete system output is displayed. This will feel like "eyeballing" AI output, but we ask the human expert to perform a light-weight

form of highlighting relevant text spans or taking free-form notes. These are used as input for nugget formation as shown in Figure 1.

Since this grounding can also be a potential source of bias, it is important to have the expert inspect system responses in a random order and to anonymize the system names. This avoids that humans prefer content in a system that is a favored candidate for deployment. The order consideration should be tracked and carefully considered in a quality control step.

We suggest to have the human expert choose whether highlighting text spans and/or free-text is more appropriate in aiding the nugget curation process. During curation, the expert also assigns each nugget to the must-have, should-have, or avoid category introduced above.

3.2 Nugget Canonicalization Support

Finally, when teams of human experts work together on creating nuggets, there tends to be variation due to different styles of writing. For AI to offer matching support, it is best if all nuggets are somewhat similarly phrased and if the nugget formulation works well with the matching prompt.

Based on human artifacts, such as highlighted text spans and free-text notes, the AI can assist the canonical phrasing of the nuggets, and ensure that nuggets are atomic and self-contained and make unambiguous reference to the query. This process is depicted in Figure 2.

It is important that the AI's role is restricted to formalization. The human expert decides what is essential information; the LLM merely helps express it as a verifiable question. Hence the AI is not permitted to propose what is relevant.

While it is possible for an LLM to propose nuggets without human input [5, 11, 16, 22], it significantly weakens the guardrail against circularity [6] and limits whether the human experts are genuinely accountable for the resulting decisions.

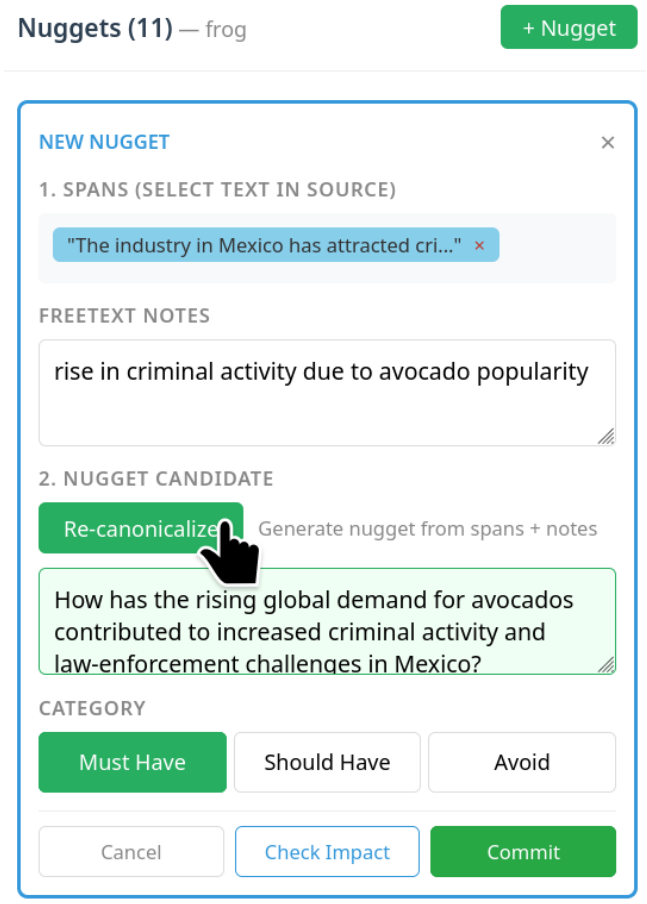

Figure 2: Nugget canonicalization support. After clicking "Canonicalize", the LLM converts selected spans, free-text, and the task description into one nugget. In this prototype, the alignment prompt expects an open-ended question such as: "How has the rising global demand for avocados contributed to increased criminal activity and law-enforcement challenges in Mexico?". The human expert can further edit the proposed phrasing.

3.3 Nugget Impact Feedback

How a particular nugget is phrased has consequences of how well the AI is able to find all the relevant matches in system responses. If the formulation is too specific, no matches will be found, but when it is too generic, the nugget will not distinguish quality differences among the best systems. To help the human expert choose the appropriate formulation, we include a "Check Impact" feature: As soon as the expert is phrasing a nugget, the AI will identify quotes of system output that would match the nugget phrasing according to the nugget-matching prompt.

As demonstrated in Figure 3, quotes of matching system responses are displayed along with the nugget coverage grade (5 for perfect coverage, 1 for only topical references). A click on the quote will display the quote in context of the system response. For nuggets that are grounded with highlighted spans in a particular system output, the user will receive feedback on whether the highlighted text would be matched by the nugget alignment prompt.

Based on inspection of the results and match statistics, the human expert is invited to change the formulation of the nugget to obtain the intended coverage breadth and specificity.

3.4 Quality Control

After nugget matching, the quality-control view lets the human expert inspect whether the resulting leaderboard agrees with their informed judgment of the systems. Rather than treating the scores as final, the interface exposes the effects of the current nugget bank: experts can adjust category weights, select subsets of related nuggets, use solo mode to isolate the impact of individual nuggets, and check whether "avoid" nuggets penalize responses as intended.

The goal of this stage is to ensure that the nugget bank reveals meaningful quality differences among the strongest systems. The expert should also confirm that low-ranked systems genuinely omit useful information, rather than being penalized because the nugget bank does not yet include their relevant content.

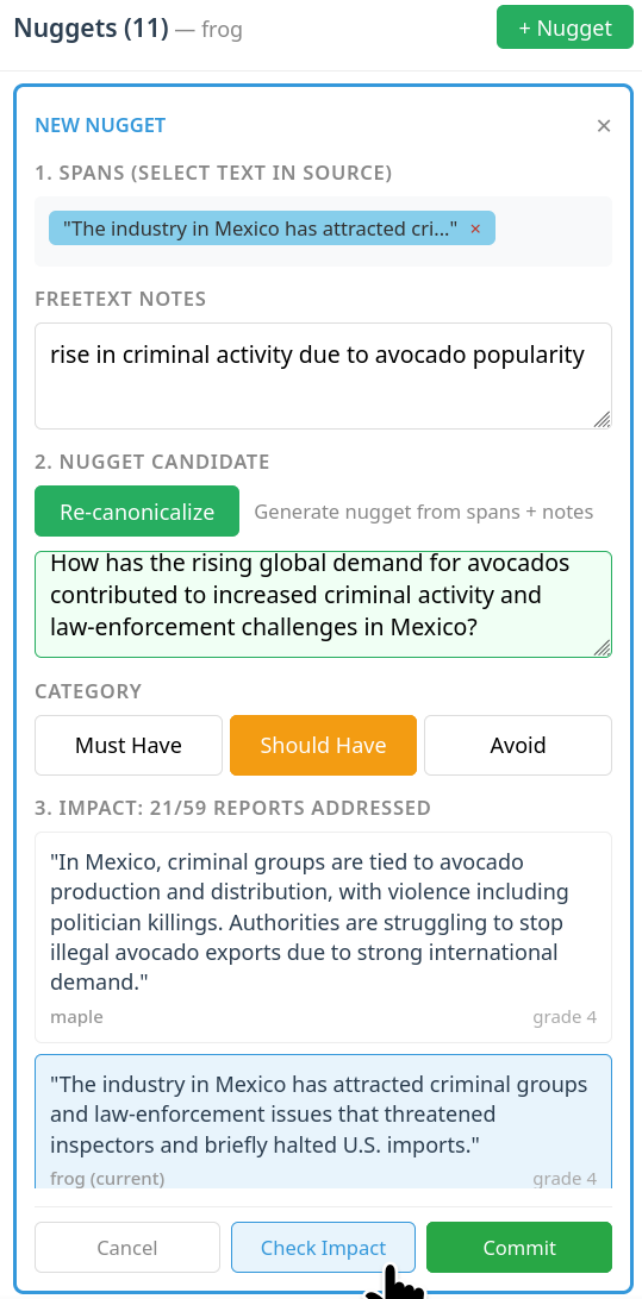

Figure 3: Check Nugget Impact. After clicking Check Impact, the interface shows in how many system outputs this nugget was located, along with supporting quotes. Here the system "maple" (grade 4) shows: "In Mexico, criminal groups are tied to avocado production and distribution, with violence including politician killings…". The annotator sees exactly how this nugget would affect system quality measurements. Here the user receives confirmation that the system response that inspired the nugget is also captured before committing the nugget.

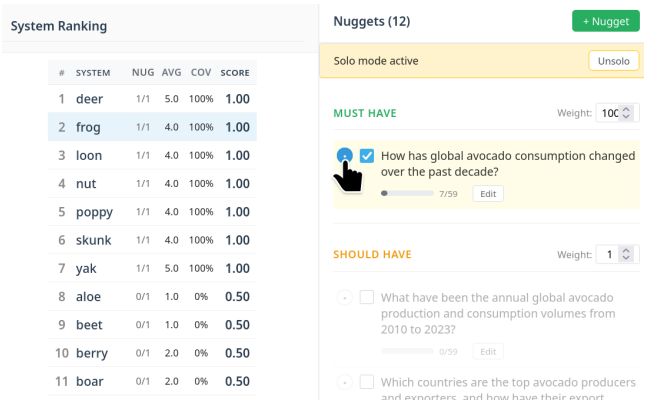

Figure 4: Inspecting nugget coverage and its impact on system ranking. The interface lets experts assess how the nugget bank affects rankings both collectively and in isolation, helping them confirm that the results align with their informed impression of system quality.

Across these steps, AI support remains downstream of a human-authored artifact, preserving human accountability while improving consistency and providing impact feedback.

4 Avoiding Evaluation Tropes

The prototype addresses several recurring failure modes in LLM evaluation by assigning each risky decision point to the human expert or an explicit inspection step.

Avoiding Anchoring (Rubber-Stamp Effect). The human identifies relevant information before seeing any LLM suggestion. Canonicalization only formalizes a human-authored artifact, so the human is not asked to accept or reject an AI proposal before forming their own judgment.

Avoiding Black-Box Labeling. Rather than asking humans for a single holistic quality score, the workflow asks them to create concrete, checkable nuggets. This makes the evaluation criteria visible as a set of nuggets instead of leaving them implicit in an opaque label. The process therefore prevents the human expert from merely saying "looks good to me" without specifying what is good or bad.

Avoiding Criteria Drift. The impact feedback preview reveals immediately when a nugget produces unexpected grades. The human discovers interpretation issues during creation, not after annotating an entire test set. The QC phase provides aggregate diagnostics, such as how many system responses fail to cover any nugget and how often each nugget is addressed across all systems.

Enabling Human Accountability. The LLM cannot unilaterally introduce evaluation criteria: every nugget exists because a human selected, wrote, categorized, or edited it. This makes accountability inspectable at the level of individual nugget phrasing decisions, than only at the level of aggregate scores which are difficult to interpret.

Maintaining Genuine Oversight. Because system quality measurements are tied to individual nuggets and supporting matches, they can be inspected, audited, and revisited when systems or standards drift. The exported nugget bank therefore serves as a human-grounded reference that can be reused for method development, LLM evaluation, and deployment observability.

5 Related Work

Nugget-based evaluation. Nugget-based evaluation has a two-decade history in information retrieval, originating at TREC [29]. Nenkova and Passonneau [19], Voorhees [29] introduced nugget pyramids for question answering evaluation. Lin and Demner-Fushman [17] explored automated matching. Pavlu et al. [21] use nuggets to evaluate information retrieval systems. Sander and Dietz [24] use multiple-choice questions with a Q/A system to evaluate compound responses of retrieved paragraphs.

Early nugget evaluations required manual matching by NIST assessors, making them expensive and difficult to scale. This work-intensive matching step hindered adoption despite the conceptual advantages of nugget-based assessment. Modern NLP and LLM capabilities make it feasible to automate the matching step while preserving human control over what to match [4, 11, 22, 24, 30].

LLM-as-a-Judge. Faggioli et al. [9], Zheng et al. [31] formalize the pattern of using LLMs for evaluation, but already discuss a range of issues. Subsequent work documented biases including position bias [27], leniency bias [10, 23], and self-preferential bias [18]. Dietz et al. [8] cataloged 14 failure modes in LLM evaluation, including the Rubber-Stamp Effect and Black-Box Labeling addressed here.

Human-AI Collaboration. Agudo et al. [1] study anchoring effects in AI-assisted labeling. Shankar et al. [26] documented criteria drift in annotation tasks. Related concerns appear in active learning and semi-supervised learning, where model-selected examples, miscalibrated confidence, and low-quality labels can undermine the value of additional human or unlabeled data [3, 13, 14, 20, 25].

6 Conclusion

The question is not whether to include humans in LLM evaluation, but how to include them effectively. The standard approaches, verify-and-correct or independent labeling, either anchor humans to machine judgments or leave them unsupported in high-variance tasks.

This prototype demonstrates a different division of labor. Humans contribute where they excel (identifying what matters) while LLMs handle what they do well (high-volume linguistic matching). The three-phase workflow supports the creation of nuggets, quality control, and to observe evaluation results. Exported nugget banks enable reproducible, auditable evaluation in any LLM evaluation or observability tool chain.

The fix for circular LLM-as-a-Judge evaluation is not a better prompt. It is a better division of labor. Three design principles readily translate to other domains where humans and AI jointly create evaluation criteria:

Human Initiative, LLM Assistance. Put the human action before the machine suggestion. In other domains, this means asking experts to articulate their intent, evidence, constraint, or concern before using AI to clean up wording, normalize format, or apply the criterion at scale.

Impact Feedback Before Commitment. Show experts the downstream effect of their decisions while they can still revise them. Natural-language criteria may seem clear in isolation, but their consequences only become visible when applied to real cases; previewing those consequences helps experts refine criteria before they become part of an evaluation pipeline.

Narrow, Verifiable LLM Tasks. Assign the LLM tasks that can be checked directly, such as rewriting a human-authored artifact into a canonical form or matching that artifact against candidate outputs. Avoid asking the LLM to make open-ended normative decisions that should remain human responsibilities.

These principles suggest a broader alternative to fully automated evaluation: reserve intellectually demanding choices for accountable human experts, and use AI to make those choices easier to express, apply, inspect, and repeat.

References

[1] Ujué Agudo, Karlos G Liberal, Miren Arrese, and Helena Matute. 2024. The impact of AI errors in a human-in-the-loop process. *Cognitive Research: Principles and Implications* 9, 1 (2024), 1.

[2] Charles L. A. Clarke and Laura Dietz. 2025. LLM-based relevance assessment still can't replace human relevance assessment. In *EVIA 2025: Proceedings of the Tenth International Workshop on Evaluating Information Access (EVIA 2025), a Satellite Workshop of the NTCIR-18 Conference, June 10-13, 2025, Tokyo, Japan*. 1–5. doi:10.20736/0002002105

[3] Fabio Gagliardi Cozman, Ira Cohen, and M Cirelo. 2002. Unlabeled Data Can Degrade Classification Performance of Generative Classifiers.. In *FLAIRS*. 327–331.

[4] Laura Dietz. 2024. A workbench for autograding retrieve/generate systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1963–1972.

[5] Laura Dietz, Naghmeh Farzi, Eugene Yang, and Dawn Lawrie. 2026. Too Many Questions: Deriving Concise and Effective Nugget Banks. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '26)* (July 20–24, 2026). ACM, Melbourne, VIC, Australia.

[6] Laura Dietz, Bryan Li, Eugene Yang, Dawn Lawrie, William Walden, and James Mayfield. 2026. Insider Knowledge: How Much Can RAG Systems Gain from Evaluation Secrets?. In *Proceedings of the 48th European Conference on Information Retrieval (ECIR 2026)*. arXiv:2601.13227.

[7] Laura Dietz, Oleg Zendel, Peter Bailey, Charles Clarke, Ellese Cotterill, Jeff Dalton, Faegheh Hasibi, Mark Sanderson, and Nick Craswell. 2025. Principles and Guidelines for the Use of LLM Judges. In *Proceedings of the 11th ACM SIGIR / The 15th International Conference on Innovative Concepts and Theories in Information Retrieval*.

[8] Laura Dietz, Oleg Zendel, Peter Bailey, Charles L. A. Clarke, Ellese Cotterill, Jeff Dalton, Faegheh Hasibi, Mark Sanderson, and Nick Craswell. 2025. Principles and Guidelines for the Use of LLM Judges. In *Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR '25)*. doi:10.1145/3731120.3744588

[9] Guglielmo Faggioli, Laura Dietz, Charles L. A. Clarke, Gianluca Demartini, Matthias Hagen, Claudia Hauff, Noriko Kando, Evangelos Kanoulas, Martin Potthast, Benno Stein, et al. 2023. Perspectives on Large Language Models for Relevance Judgment. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval*. 39–50.

[10] Naghmeh Farzi and Laura Dietz. 2024. Exam++: Llm-based answerability metrics for ir evaluation. In *Proceedings of LLM4Eval: The First Workshop on Large Language Models for Evaluation in Information Retrieval*.

[11] Naghmeh Farzi and Laura Dietz. 2024. Pencils Down! Automatic Rubric-based Evaluation of Retrieve/Generate Systems. In *Proceedings of the 2024 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '24)*. doi:10.1145/3664190.3672511

[12] Raymond Fok and Daniel S Weld. 2024. In search of verifiability: Explanations rarely enable complementary performance in AI-advised decision making. *AI Magazine* 45, 3 (2024), 317–332.

[13] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*. PMLR, 1321–1330.

[14] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language Models (Mostly) Know What They Know. *arXiv preprint arXiv:2207.05221* (2022).

[15] Mamoru Komachi, Taku Kudo, Masashi Shimbo, and Yuji Matsumoto. 2008. Graph-based Analysis of Semantic Drift in Espresso-like Bootstrapping Algorithms. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Honolulu, Hawaii, 1011–1020.

[16] Bryan Li, William Walden, Yu Hou, Gabrielle Kaili-May Liu, Dawn Lawrie, James Mayfield, Eugene Yang, Chris Callison-Burch, and Laura Dietz. 2026. DoGMaTiQ: Automated Generation of Question-and-Answer Nuggets for Report Evaluation. In *Proceedings of the 2026 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '26)*. ACM, Melbourne, VIC, Australia.

[17] Jimmy Lin and Dina Demner-Fushman. 2006. Will pyramids built of nuggets topple over?. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*. 383–390.

[18] Yiqi Liu, Nafise Sadat Moosavi, and Chenghua Lin. 2024. LLMs as Narcissistic Evaluators: When Ego Inflates Evaluation Scores. In *Findings of the Association for Computational Linguistics (ACL) 2024*. https://aclanthology.org/2024.findings-acl.753/ Investigates bias in LLM-based evaluation metrics favoring their own outputs.

[19] Ani Nenkova and Rebecca J Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of the human language technology conference of the north american chapter of the association for computational linguistics: Hlt-naacl 2004*. 145–152.

[20] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. 2018. Realistic evaluation of deep semi-supervised learning algorithms. *Advances in neural information processing systems* 31 (2018).

[21] Virgil Pavlu, Shahzad Rajput, Peter B. Golbus, and Javed A. Aslam. 2012. IR System Evaluation Using Nugget-Based Test Collections. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining (WSDM 2012)*. ACM, Seattle, Washington, 393–402.

[22] Ronak Pradeep, Nandan Thakur, Shivani Upadhyay, Daniel Campos, Nick Craswell, and Jimmy Lin. 2024. Initial Nugget Evaluation Results for the TREC 2024 RAG Track with the AutoNuggetizer Framework. (2024). https://arxiv.org/abs/2411.09607 ArXiv preprint.

[23] Ronak Pradeep, Nandan Thakur, Shivani Upadhyay, Daniel Campos, Nick Craswell, Ian Soboroff, Hoa Trang Dang, and Jimmy Lin. 2025. The Great Nugget Recall: Automating Fact Extraction and RAG Evaluation with Large Language Models. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 180–190.

[24] David P Sander and Laura Dietz. 2021. EXAM: How to Evaluate Retrieve-and-Generate Systems for Users Who Do Not (Yet) Know What They Want.. In *DESIRES*. 136–146.

[25] Burr Settles. 2009. *Active learning literature survey*. Technical Report.

[26] Shreya Shankar, JD Zamfirescu-Pereira, Björn Hartmann, Aditya Parameswaran, and Ian Arawjo. 2024. Who validates the validators? aligning llm-assisted evaluation of llm outputs with human preferences. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. 1–14.

[27] Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. 2025. Judging the judges: A systematic study of position bias in llm-as-a-judge. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*. 292–314.

[28] Amos Tversky and Daniel Kahneman. 1974. Judgment under Uncertainty: Heuristics and Biases: Biases in judgments reveal some heuristics of thinking under uncertainty. *science* 185, 4157 (1974), 1124–1131.

[29] Ellen M. Voorhees. 2003. Overview of the TREC 2003 Question Answering Track. In *Proceedings of the Twelfth Text REtrieval Conference (TREC 2003)*. NIST, Gaithersburg, Maryland.

[30] William Walden, Marc Mason, Orion Weller, Laura Dietz, John Conroy, Neil Molino, Hannah Recknor, Bryan Li, Gabrielle Kaili-May Liu, Yu Hou, et al. 2026. Auto-argue: Llm-based report generation evaluation. In *SIGIR*.

[31] 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, and Ion Stoica. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*. 46595–46623.

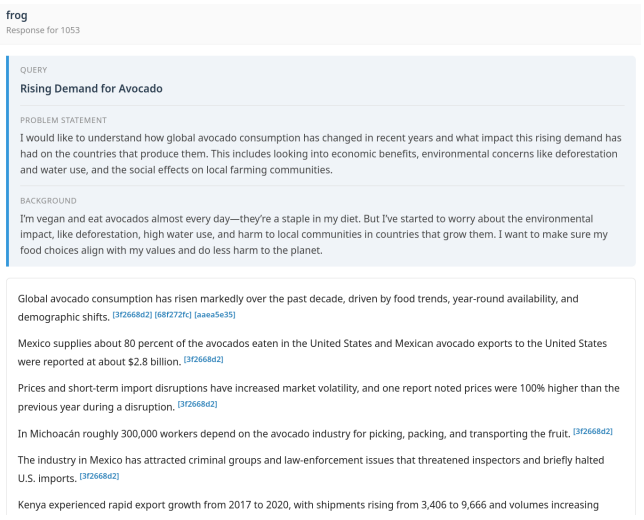

Figure 5: Step 1: Human reads the query and report. The annotator sees the query "Rising Demand for Avocado" with its problem statement and background context. Below, the system response describes global avocado trends, Mexico's market share, and various impacts. The human reads and forms their own understanding of what information matters. No LLM has been invoked.

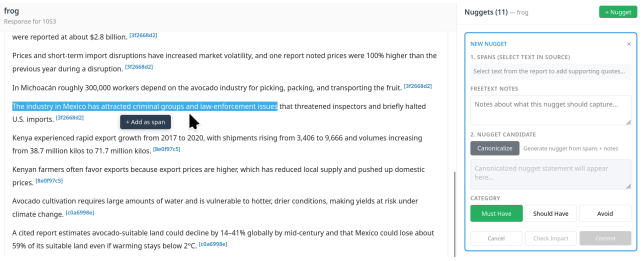

Figure 6: Step 2: Human selects text. The annotator highlights text: "The industry in Mexico has attracted criminal groups and law-enforcement issues." A popup appears offering "+ Add as span." The draft card opens on the right. The LLM has still not been involved. The human has already identified what they consider important.

A Appendix: Walkthrough

This appendix demonstrates the prototype through three scenarios that illustrate the key design principles.

A.1 Human Goes First, AI Supports

This walkthrough shows how the interface ensures the human forms their judgment before any LLM involvement.

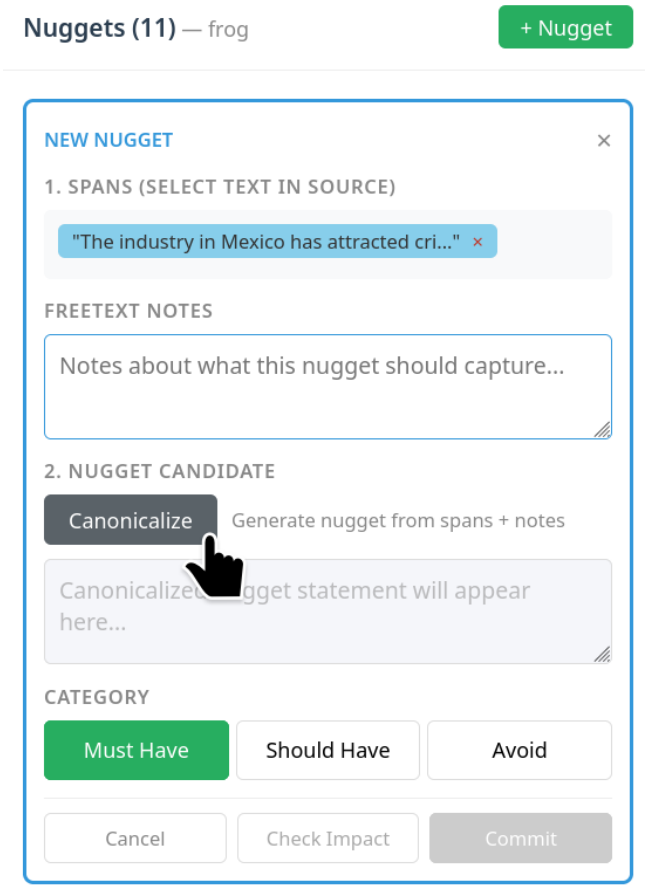

Figure 7: Step 3: Span added, ready for notes. The draft card shows the selected span as a chip. The Canonicalize button is visible but not yet clicked. At this point, the human's conceptual judgment is already formed.

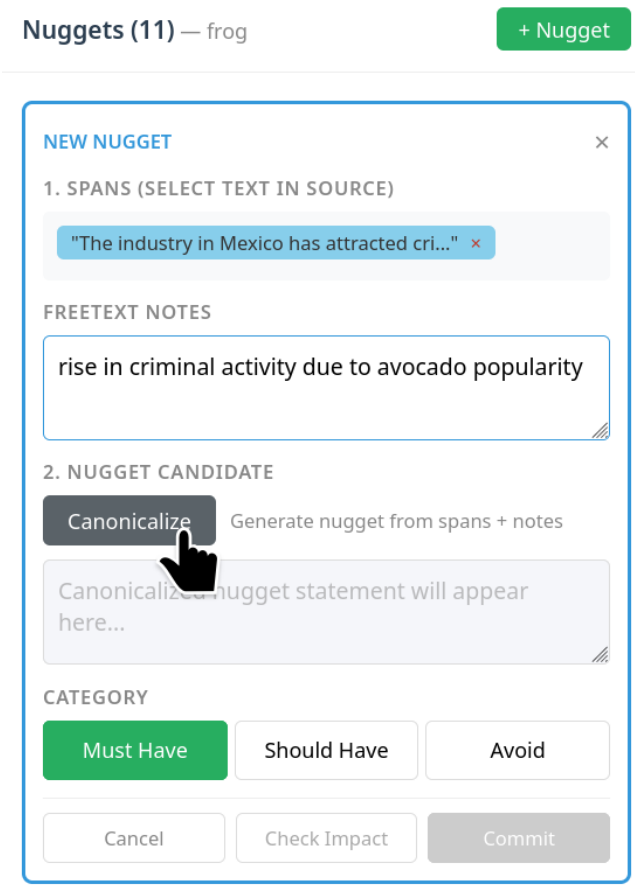

Figure 8: Step 4: Human adds context. The annotator types free-text notes: "rise in criminal activity due to avocado popularity." These notes describe their intent before any machine involvement.

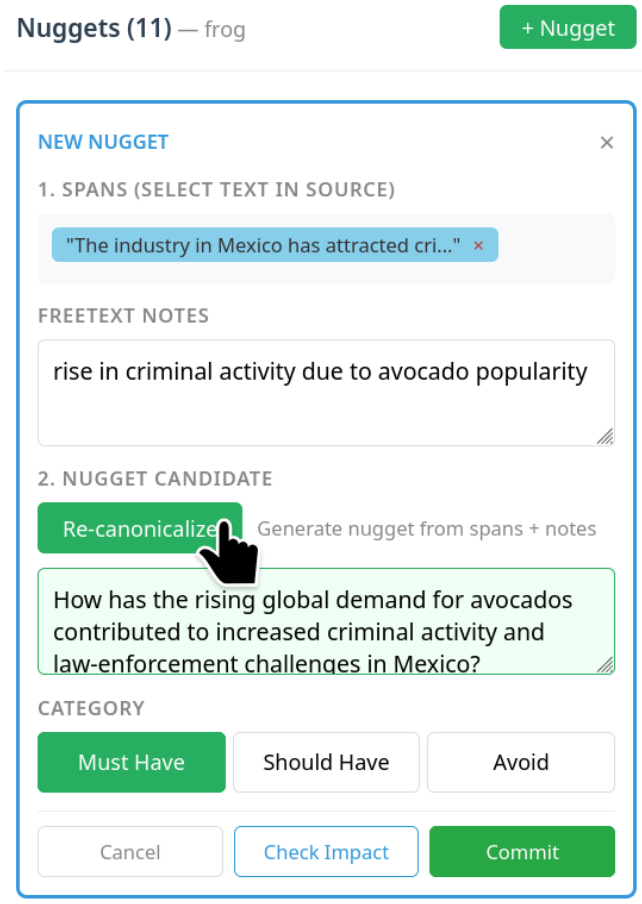

Figure 9: Step 5: LLM formalizes. After clicking Canonicalize, the LLM generates: "How has the rising global demand for avocados contributed to increased criminal activity and law-enforcement challenges in Mexico?" The LLM's role is formalization, not proposal. The human decided this information matters; the LLM helped express it as a verifiable question.

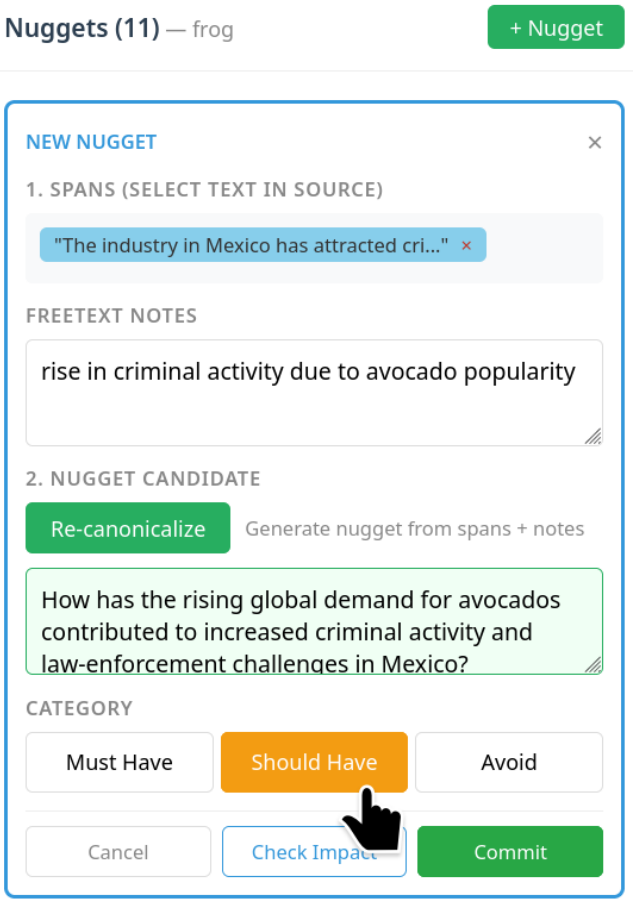

Figure 10: Step 6: Human chooses category. The annotator selects "Should Have" as the category. The importance judgment is entirely human.

A.2 Feedback Loop via Check Impact

This walkthrough shows how real-time feedback helps the annotator refine nuggets before committing.

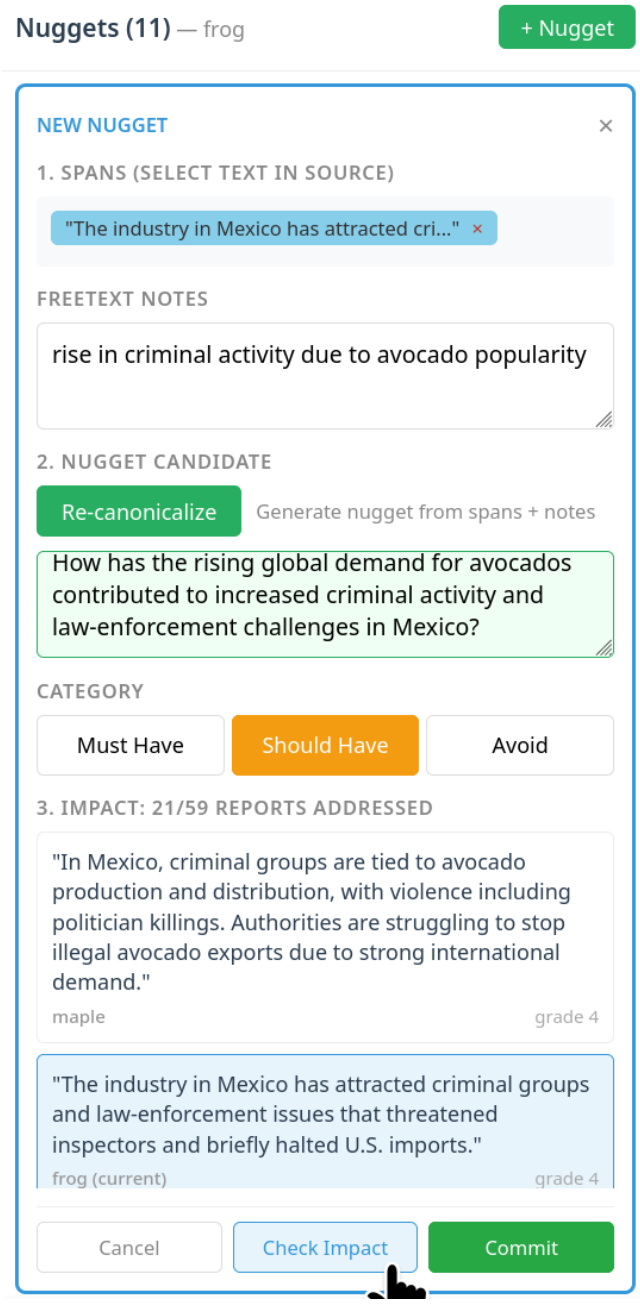

Figure 11: Step 1: Impact results with quotes. After clicking Check Impact, the interface shows "21/59 REPORTS ADDRESSED" with supporting quotes. System "maple" (grade 4) shows: "In Mexico, criminal groups are tied to avocado production and distribution, with violence including politician killings…" The annotator sees exactly how this nugget grades and why.

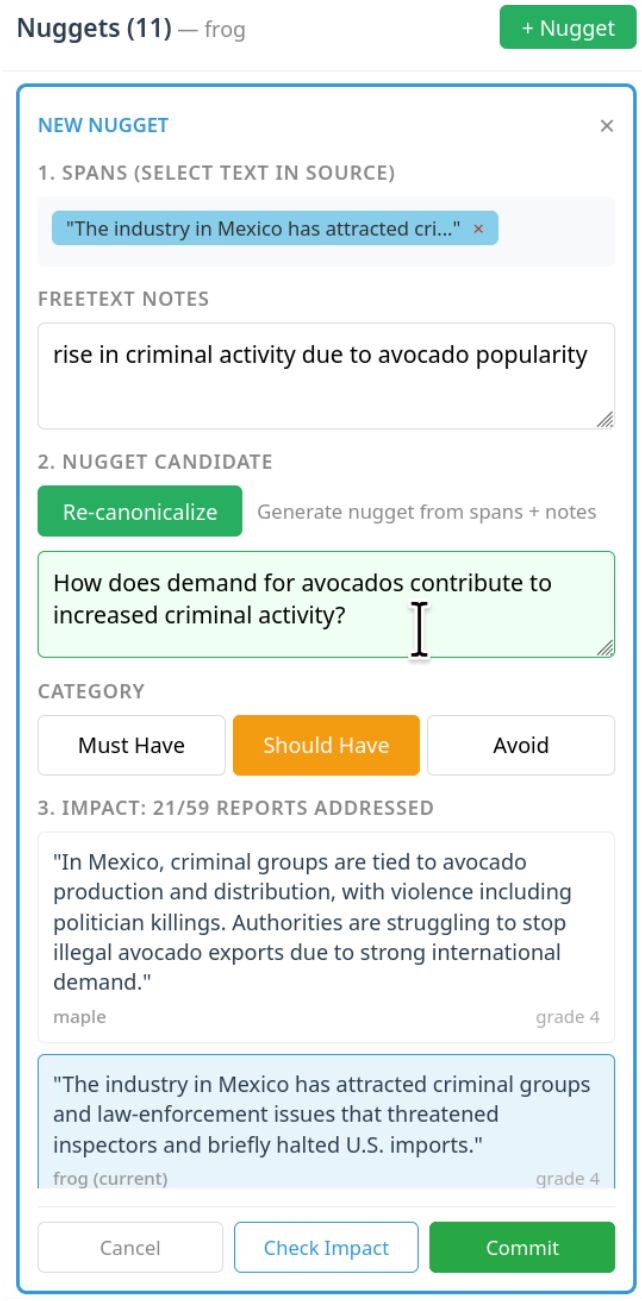

Figure 12: Step 2: Refining based on feedback. Based on the preview, the annotator edits the nugget text to be more concise: "How does demand for avocados contribute to increased criminal activity?" This demonstrates the feedback loop: the annotator saw how the nugget performed and refined it.

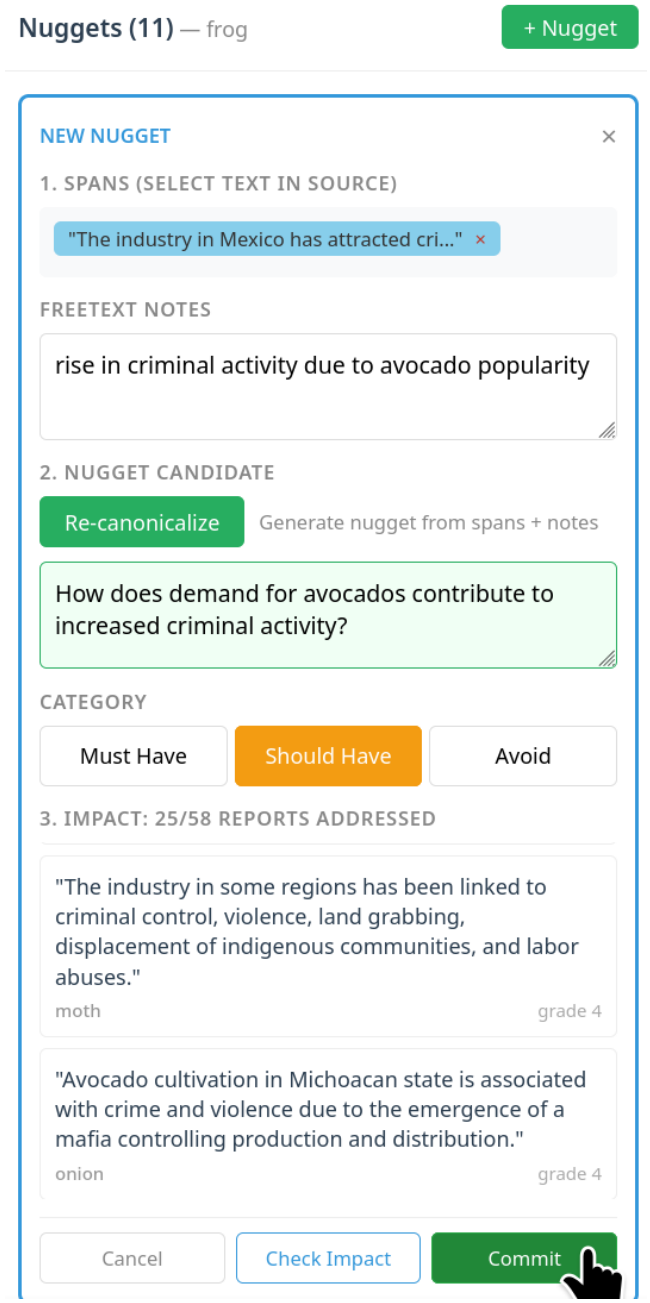

Figure 13: Step 3: Final verification. The updated nugget shows "25/58 REPORTS ADDRESSED" with quotes from additional systems. The annotator clicks Commit, satisfied that the nugget discriminates effectively.

A.3 QC Phase for Improving Nugget Bank Quality

This walkthrough shows how the annotator tunes and validates the nugget bank.

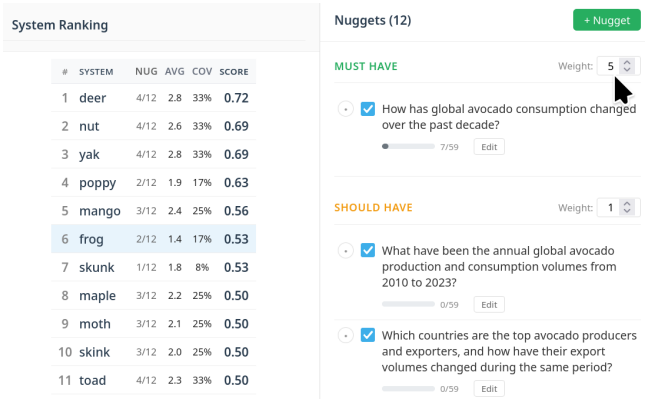

Figure 14: Step 1: QC phase with weight controls. The QC phase shows the System Ranking table with columns for rank, system, nuggets satisfied (NUG), average grade (AVG), coverage (COV), and weighted score (SCORE). Category weights are adjustable (Must Have: 5, Should Have: 1).

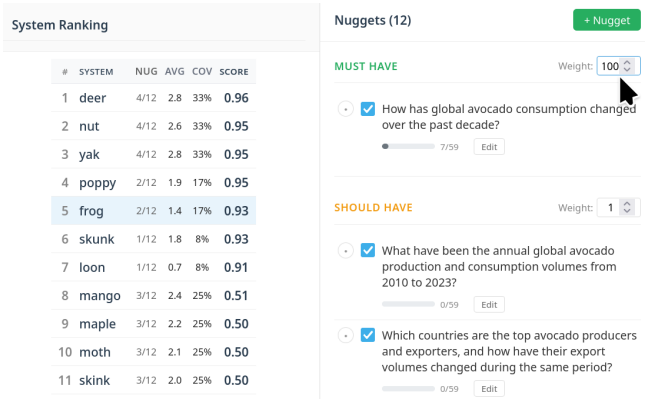

Figure 15: Step 2: Weight adjustment changes rankings. Increasing Must Have weight to 100 causes immediate ranking changes. Scores compress and reorder. This reveals how sensitive the ranking is to category weights.

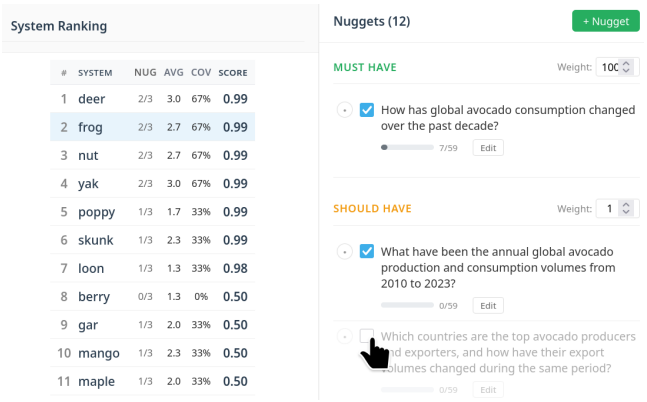

Figure 16: Step 3: Disable a nugget. Unchecking a nugget removes it from scoring. The NUG column changes from "4/12" to "2/3." This allows testing whether specific nuggets contribute meaningfully.

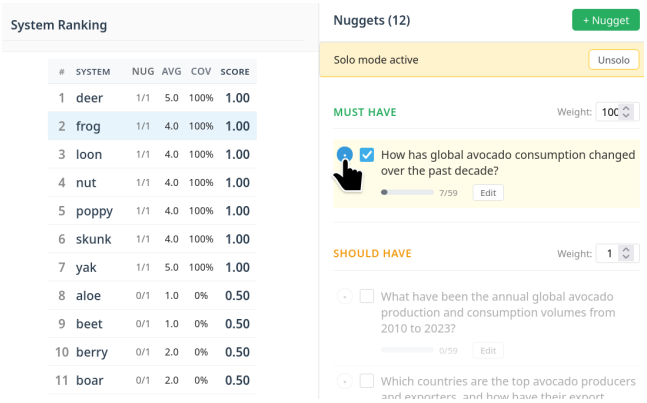

Figure 17: Step 4: Solo mode. Solo mode isolates a single nugget. Systems split cleanly: those addressing the nugget score 1.00; others score 0.50. This reveals the nugget's discriminative power.

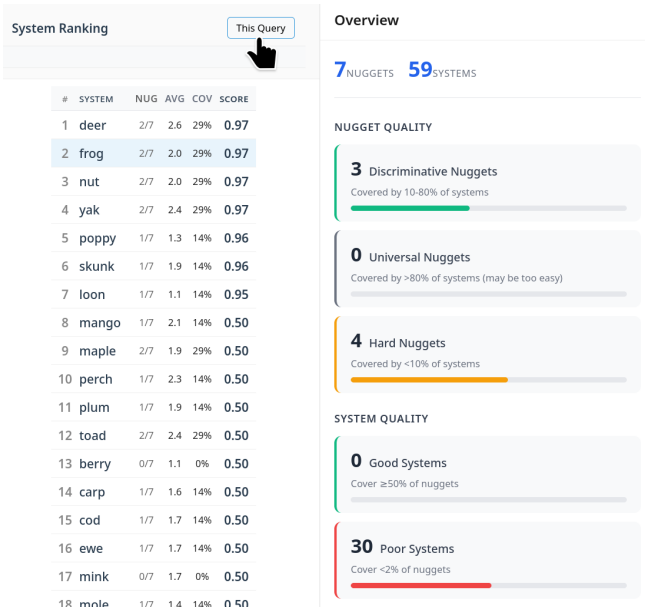

Figure 18: Step 5: Observe phase diagnostics. The Observe phase shows aggregate statistics: 7 Nuggets, 59 Systems. Nugget Quality: 3 Discriminative (10–80% coverage), 0 Universal, 4 Hard. System Quality: 0 Good Systems, 30 Poor Systems. The 4 Hard nuggets suggest the bank may be too strict.

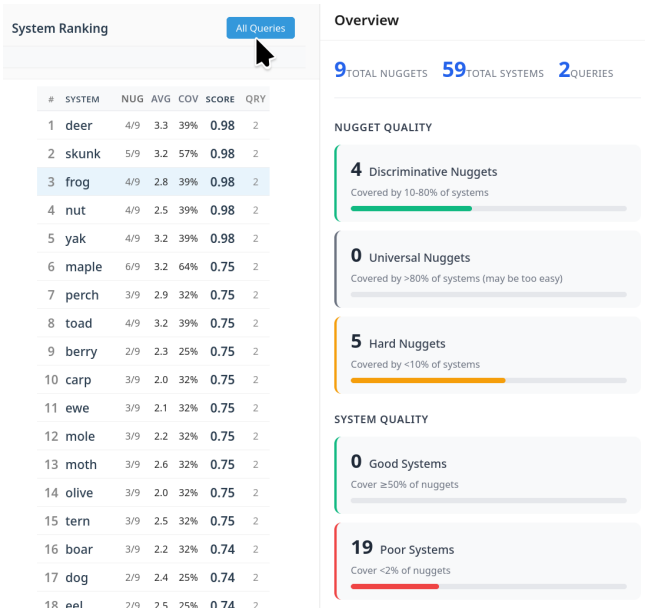

Figure 19: Step 6: Cross-query view. "All Queries" shows macro-averaged statistics across 2 queries: 9 Total Nuggets, 4 Discriminative, 5 Hard. A QRY column shows per-system query coverage. This reveals which systems perform consistently.

A.4 Summary of Principles Demonstrated

Walkthrough	Principle	How Demonstrated
A.1	Human initiative	Selection and notes before canonicalization
A.1	LLM assists, does not propose	Canonicalize formalizes human-identified information
A.1	No anchoring	Human judgment formed before seeing LLM output
A.2	Feedback before commit	Check Impact reveals grades and quotes instantly
A.2	Avoiding criteria drift	Refinement happens during creation
A.2	Verifiable matching	Quotes show exactly why grades were assigned
A.3	Interactive calibration	Real-time weight adjustment
A.3	Hypothesis testing	Disable and solo individual nuggets
A.3	Aggregate diagnostics	Discriminative, universal, and hard nugget counts

Table 1: Walkthrough principles