# Critical Evaluation of Knowledge Graph-powered Question Answering Systems to Enable Responsible AI for Disaster Management

**Gayatri Ramchandra Vaidya, Yasas Senarath, Hemant Purohit**
Humanitarian Informatics Lab, George Mason University, Fairfax, VA, USA
{gvaidya2, ywijesu, hpurohit}@gmu.edu

## Abstract

When a disaster strikes, question-answering (QA) systems promise to turn scattered satellite, infrastructure, and hazard impact data into actionable answers for coordinators working under time pressure. But how critically should we trust those answers, even if grounded in the background knowledge? We present RISE, a prototype knowledge-graph (KG)-powered QA system built on data from the June 2026 Venezuela M7.5 earthquake, and subject it to a critical evaluation. Two complementary studies run on the same 30 questions: a stage-wise trace records how each question moves through the pipeline, while an independent LLM-as-a-judge assessment scores the final answer on five language-quality criteria. Taken separately, both studies suggest a system that works: 87% intent accuracy, every question answered, and judge scores averaging above 4 out of 5. Albeit together, they reveal a different picture. Intent misclassifications trace to a linguistic ambiguity no keyword-based classifier can resolve: the same word serves as context in one question and as the information need in another, and lexical matching cannot tell them apart. High scores on grounding and appropriateness do not distinguish a correct answer from a fluent, well-cited answer to the wrong question, because the rubric scores the text against its evidence rather than against the question. Both findings share a common lesson for responsible AI in disaster management to design and develop effective support systems: fluent, well-grounded language by the AI agent is not evidence of a correct answer, and evaluating these systems on their text alone is not enough.

## 1 Introduction

Humanitarian coordinators, emergency responders, and managers often face a familiar problem: the data to make decisions exists but is scattered across different satellites, hazard zones, and pre-disaster infrastructure records, each in a different format with nothing linking them together. For example, which roads are blocked, which health facilities are damaged, and which municipalities were hit hardest? Integrating these heterogeneous sources into a knowledge graph and letting a user ask questions in plain language has been proposed as a solution. Purohit et al. (2019) proposed a SPARQL-based querying application for disaster-related question answering, noting the substantial cost of manual semantic integration. Xiao et al. (2026) show that standard text-to-SQL methods transfer poorly to disaster analytics and address this with an expert-curated causal knowledge graph between the user's question and the database. But neither examines whether a system that retrieves and routes to the correct data can still produce an answer whose language misleads the person reading it.

This distinction matters for responsible AI in disaster management. Existing RAG systems ground their output in retrieved evidence, and this grounding is assumed to make answers more trustworthy than those generated from the model's internal knowledge alone (Zhou et al., 2024). But recent work questions this assumption. An et al. (2026) separate *lexical grounding* (whether evidence is cited) from *semantic alignment* (whether it is interpreted correctly), and show that explanations can appear grounded even when their interpretation weakens the intended meaning. LLM-as-a-judge evaluation, where a strong model scores answers on quality dimensions, agrees with human judgments over 80% of the time (Zheng et al., 2023), but a judge shown only the answer and its evidence can assess whether the answer matches the evidence, not whether the evidence was right for the question.

What remains largely unexplored is how this gap appears in a working KG-QA pipeline built on real disaster data. Standard evaluations report execution accuracy (Yu et al., 2018; Li et al., 2023), treating a correct result set as success. Xiao et al. (2026) extend this with routing accuracy, causal-



*Proceedings of the First Workshop on Language Technologies for Crisis Preparedness and Response*, pages 126–133
November 9, 2026 ©2026 Association for Computational Linguistics

graph grounding, and multi-table composition as distinct failure tiers. But all these methods check whether the right data was retrieved and correctly cited. A query can execute successfully, return real records, and be written faithfully yet still fail to answer the actual question.

We address this gap through a critical evaluation of RISE, an AI-powered Recovery Intelligence and Support Engine: a prototype KG-powered QA system built on data from the June 2026 Venezuela earthquake. We ask how critically coordinators should trust its answers, even when those answers are grounded in background knowledge. Our contributions are threefold. First, we present RISE, which combines independent damage-assessment sources with pre-disaster infrastructure and administrative geography, backed by a seven-stage NL-to-Cypher pipeline. Second, we introduce two complementary studies on the same 30 questions: a stage-wise trace of how each question moves through the pipeline, and an independent LLM-as-a-judge assessment of the final answer on five language-quality criteria. Third, we show that the two studies, read together, reveal failures that neither shows alone: intent errors that trace to a linguistic ambiguity no keyword-based classifier can resolve, and high grounding scores that cannot distinguish a correct answer from a fluent, well-cited answer to the wrong question.

Together, these findings carry an important implication for responsible AI in disaster management and for anyone designing decision-support systems: fluent, well-grounded language from an AI agent is not evidence of a correct answer. Evaluating such systems on their text alone is therefore not enough; we must also examine how each answer was produced.

## 2 Related Work

**Knowledge graphs for disaster response.** Purohit et al. (2019) introduced DisasterKG, an RDF integration of heterogeneous disaster data, noting the high cost of manual semantic integration. Xiao et al. (2026) show that standard text-to-SQL pipelines transfer poorly to disaster analytics and address this with an expert-curated knowledge graph between the user's question and the database. Neither examines whether a system that retrieves and routes correctly can still generate misleading output.

**Text-to-query systems and evaluation.** Yu et al. (2018) evaluate natural-language-to-database queries by execution accuracy, and Li et al. (2023) extend this to large real-world databases, judging a query by its result set regardless of surface form. Execution accuracy checks whether the retrieved data is right; it does not check whether the answer built from that data is right for the question. For KG-grounded QA, Pusch and Conrad (2025) show that validating and correcting LLM-generated queries against the schema before execution reduces hallucination. RISE follows this validation approach; our contribution examines what happens after validation succeeds, when a query can answer a different question while remaining fluent and grounded.

**LLM-as-a-judge and the grounding illusion.** Zheng et al. (2023) show that strong LLM judges agree with human preferences over 80% of the time, which has made LLM-as-a-judge a common evaluation method. An et al. (2026) introduce VEXA, a controlled testbed separating *lexical grounding* (whether evidence is cited) from *semantic alignment* (whether it is interpreted correctly), and show that explanations can appear grounded even when their interpretation weakens the intended message. We demonstrate that the same problem can arise in a working system built on real disaster data: a query may retrieve valid records and produce a fluent response while still answering a different question from the one originally asked.

## 3 RISE: System Architecture

RISE is designed to answer natural-language questions concerning the June 2026 Venezuela earthquake by translating user queries into Cypher, retrieving relevant information from a Neo4j knowledge graph, and generating grounded natural-language responses from the returned records. The same knowledge graph also supports an interactive map and chatbot interface, enabling spatial visualization of retrieved information for situational awareness. The system consists of three principal components: 1) a Knowledge Graph, 2) an NL-to-Cypher Query Pipeline, and 3) an Interactive User Interface. Figure 1 shows the RISE query pipeline. The system is guided by the following design principles:

*Source Provenance* is preserved by storing each damage assessment independently, since estimates derived from different modalities and methods may disagree substantially. This allows agreement and disagreement across sources to remain directly

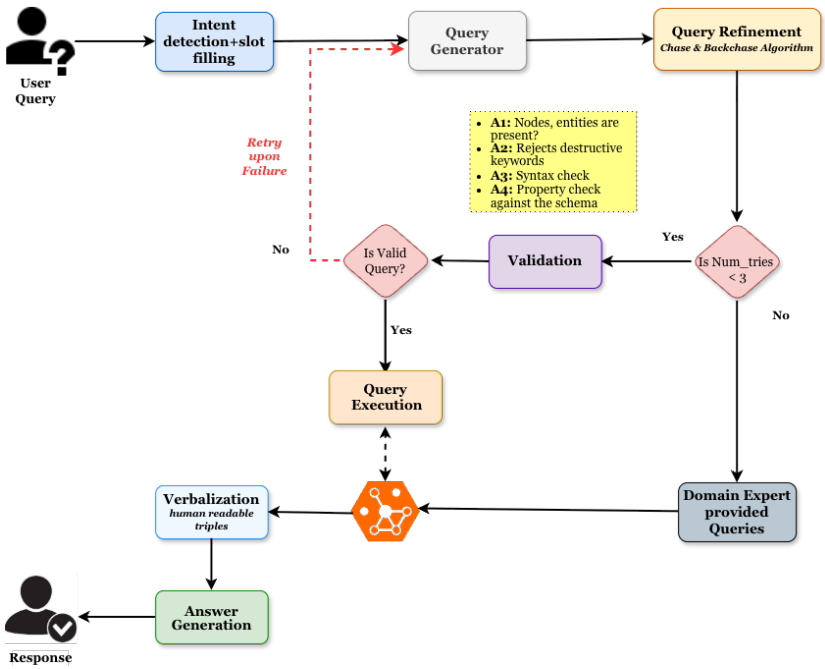



Figure 1: RISE query pipeline architecture.

queryable.

*Scope Enforcement* limits the knowledge graph to physical damage and location information derived from satellite imagery and OpenStreetMap data. Information on casualties, displacement, shelters, utilities, and agency responsibility is explicitly excluded and enforced during answer generation.

*Local Inference* is supported through Ollama, ensuring that user queries and retrieved information remain on the host system. This design supports operation under limited connectivity and data-residency constraints, while restricting the system to locally deployable models.

### 3.1 Knowledge Graph

A knowledge graph represents information as nodes (entities) connected by typed edges (relationships), letting questions be answered by traversing paths rather than joining tables. RISE uses Neo4j with Cypher as its query language, over four groups of information. **Building damage** consists of independent radar and optical assessments (analyst-annotated and AI-derived) stored as `DamagedBuilding` nodes, with confidence and severity kept as separate properties since radar sources report a confidence tier while optical sources report a severity grade. The **pre-earthquake baseline** includes building footprints, roads, and health and education facilities from OpenStreetMap, together with hazardous industrial sites from UNEP/OCHA, with facility subtypes preserved. **Hazard and administrative context** covers the earthquake, its aftershocks, and USGS shaking-intensity zones, alongside states and municipalities from OCHA Common Operational Datasets. Finally, **provenance** is captured through `SOURCED_FROM` edges linking every entity to its data source, while `LOCATED_IN` edges attach entities to every applicable administrative level, so municipality-level questions are answered by direct traversal rather than aggregation.

### 3.2 The NL-to-Cypher Pipeline

Each question passes through a pipeline motivated by prior work on LLM–KG question answering (Linders and Tomczak, 2025; Pusch and Conrad, 2024), adapted for post-earthquake disaster response.

**Intent Detection.** Ordered keyword rules classify the question into one of eight intents (*damage_extent*, *road_status*, *health_access*, *education_impact*, *shaking_intensity*, *earthquake_info*, *area_summary*, *out_of_scope*) and extract slots; unmatched questions fall through to a local `llama3.1` (Arora et al., 2024).

**Query Generation.** A local `qwen2.5-coder` generates a Cypher query from the intent, slots, live schema, and a small set of exemplars (Wang et al., 2026).

**Refinement.** A Chase and Backchase strategy (Wang et al., 2026) inserts missing relationship hops and removes fields not matching live property keys.

**Validation.** Four checks run before execution: node/relationship labels (A1), destructive keywords (A2), property keys (A3), and `EXPLAIN` for syntactic errors (A4). Failures return an error hint to the generator, which retries up to three times before falling back following Pusch and Conrad (2024), who show query-checking reduces hallucination in KG-grounded QA.

**Execution.** The validated query runs against Neo4j. On failure, empty results, or wrong-level rows, the system substitutes a domain-expert template.

**Fallback Queries.** Pre-authored Cypher templates keyed by intent serve as a deterministic safety net, so the system always returns an answer (Purohit et al., 2019).

**Verbalization.** Result rows are converted into subject-relation-object triples.

**Answer Generation.** Triples and the executed query are passed to a local `llama3.1` (Grattafiori et al., 2024), with a prompt that constrains the response to the provided evidence and forbids inference beyond it (e.g., inferring operational status from physical proximity, or reporting figures at a different administrative level than the retrieved rows).

### 3.3 Interface and Implementation

RISE runs on a FastAPI backend with a Leaflet map front end. Each answer returns both prose and the retrieved records, so a question about a specific area is answered in text and highlighted on the map simultaneously.

## 4 Experimental setup

We evaluate RISE through two complementary studies conducted on the same set of questions. The first is a stage wise evaluation of the query pipeline which identifies where each question is resolved and, importantly, distinguishes questions handled successfully by the generated query from those recovered through the domain-expert fallback (Section 4.2). The second is an LLM-as-a-judge evaluation of the final responses, where answers are scored along five quality dimensions (Section 4.3). These evaluations capture different aspects of system performance: the first examines how the pipeline processes a question, while the second assesses the quality of the answer ultimately presented to the user. Considering both is important because a fluent answer may be produced even when the underlying query fails, while a correctly processed query may still yield a poor verbalization. Using the same questions in both studies therefore allows us to relate pipeline behavior to answer quality at the level of individual queries.

| Intent | Questions |
|---|---|
| `damage_extent` | 6 |
| `road_status` | 4 |
| `health_access` | 4 |
| `shaking_intensity` | 4 |
| `education_impact` | 3 |
| `earthquake_info` | 3 |
| `area_summary` | 3 |
| `out_of_scope` | 3 |
| **Total** | **30** |

Table 1: Intent distribution of the gold evaluation set.

### 4.1 Evaluation Dataset

The evaluation dataset consists of 30 questions covering all eight intents supported by RISE, including three questions deliberately outside the knowledge graph's scope. The questions were generated by the authors based on inputs from a humanitarian expert on the typical decision-support needs of different stakeholders involved in disaster response and recovery. Each question is annotated with its expected intent. Table 1 summarizes the distribution across the intents.

Three of the 30 questions intentionally request information that is not represented in the knowledge graph: whether facilities are operational, where shelters are located, and whether a particular area has been surveyed. For these questions, explicitly stating that the requested information is unavailable constitutes the correct behavior rather than a failure to answer. Accordingly, the stage-wise evaluation treats `out_of_scope` as a distinct class, while the LLM-as-a-judge rubric instructs evaluators to reward appropriate refusals rather than penalizing them for lack of informativeness. Both studies use the same 30 questions and retain matching question identifiers, so their results can be compared at the level of individual questions.

This makes it possible to examine whether the route taken through the pipeline (for example, successful query generation versus fallback substitution) is reflected in the quality of the final response.

Section 4.6 uses this joint analysis to examine the relationship between pipeline behavior and answer quality.

### 4.2 Study 1: Stage-Wise Pipeline Evaluation

An end-to-end pass/fail judgement conflates two outcomes that this system needs to distinguish: a question answered because the generator wrote a correct Cypher query, and a question answered because generation failed and a hand-written fallback query covered for it. Both can produce an equally fluent sentence. Study 1 therefore instruments each of the seven pipeline stages directly, recording what the stage *did*, not only whether the run as a whole produced an answer.

**Repeated runs.** Cypher generation is performed by a 1.5B-parameter local model (Qwen2.5-coder) and is not deterministic: the same question can be generated correctly on one run and fail on the next. A single pass over the gold set would therefore understate variance and could make an unstable question look either reliably solved or reliably broken depending on which run was sampled. Each of the 30 questions is run three times, for 90 runs total, so that per-question stability can be assessed alongside per-question correctness.

**Per-stage instrumentation.** Table 2 lists what is recorded at each stage. Intent detection is scored against the gold label from Table 1 and additionally records whether the question was resolved by a deterministic keyword rule or fell through to the local LLM classifier, since these two paths have different failure characteristics. Query generation distinguishes a `dynamic` outcome (the generated query produced the returned rows) from `fallback` (generation failed or returned nothing and the hardcoded query answered instead) and `no_answer` (neither produced rows). Refinement records whether the query was modified before execution or passed through unchanged. Validation records whether the query passed its four checks (label whitelist, forbidden keywords, property-name check, an `EXPLAIN` dry-run) on the first attempt, after a retry, or not at all within the three attempts allowed.

| Stage | Recorded Outcome |
|---|---|
| Intent detection | keyword- or LLM-routed |
| Query generation | `dynamic`, `fallback`, or `no_answer` |
| Refinement | Modified before execution or unchanged |
| Validation | Passed on first attempt, after retry, or failed all attempts |

Table 2: Per-stage outcomes recorded in Study 1.

### 4.3 Study 2: LLM-as-a-Judge Evaluation

Study 1 characterizes how an answer was produced but says nothing about whether the resulting text is any good to read. Study 2 scores the 30 answers from the same gold set on five criteria, 1-5, adapted from a standard explanation-quality rubric to a knowledge-graph question-answering setting: the object being explained is the user's question rather than a flagged message, and the supporting evidence is the retrieved subgraph rather than detector features. Table 3 lists the criteria.

For the three `out_of_scope` questions identified in Section 4.1, the rubric instructs judges that a correct statement that the information is unavailable should score highly on Helpfulness and Appropriateness, so that a well-calibrated refusal is not penalized as if it were a failure to answer.

**Grounding the judge in evidence.** The LLM-judge is shown the same retrieved graph records that the answer-generating model saw when writing its response, not the answer text in isolation. Evidence Grounding and Reasoning Support are relations between a text and its supporting evidence and cannot be assessed from the text alone.

**Excluding the answer-generating model.** Llama 3.1 8B, which writes the system's answers, is excluded from the pool of candidate judges, since a model scoring its own output tends to rate it more favorably than an independent judge would.

**Judge validation protocol.** Before scoring any real answer, each candidate judge model was evaluated on a small set of constructed probe answers designed to test whether it can separate fluency from correctness: one answer that is accurate and well-grounded, one that is fluent but hallucinated, one that is vague and content-less, and one that is a correct refusal of an out-of-scope question. Two hosted models that passed validation were retained: **gpt-oss-120b** (via Groq) and **Gemini 3.1 Flash Lite**.

| Criterion | Understanding |
|---|---|
| Helpfulness | Does the answer help the reader understand what was asked? |
| Clarity | Is the answer easy to read and understand? |
| Evidence Grounding | Does the answer refer to the specific retrieved records rather than respond generically? |
| Appropriateness | Does the answer address the question at the correct geographic level? |
| Reasoning Support | Does the answer connect the retrieved evidence to its conclusion? |

Table 3: Criteria used by the LLM judges, each scored on a 1-5 scale.

| Stage | Outcome | % |
|---|---|---|
| Intent detection | Keyword-routed | 87% (78/90) |
| | LLM-routed | 13% (12/90) |
| Query generation | Dynamic | 62% (56/90) |
| | Fallback | 38% (34/90) |
| | No answer | 0% (0/90) |
| Refinement | Modified | 8% (7/90) |
| | Unchanged | 92% (83/90) |
| Validation | First try | 69% (62/90) |
| | After retry | 12% (11/90) |
| | Failed all attempts | 19% (17/90) |

Table 4: Stage-wise outcomes across 90 runs of the 30-question gold set.

### 4.4 Study 1: Stage-Wise Pipeline Results

Table 4 summarizes the 90 runs (30 questions $\times$ 3 repeats) described in Section 4.2.

**No question is left unanswered.** All 90 runs produced an answer. An earlier pipeline iteration left a quarter of runs empty, traced to few-shot exemplars that referenced non-existent graph properties (`adm1_name`, `cop_id`); removing them closed the gap.

**Fallback has two distinct causes.** Of the 34 fallback runs, 17 never produced a query that passed validation, while the other 17 passed validation but returned zero rows on execution. Half of "fallback" is thus not a generation failure but a valid query with no matching data, a distinction that points to different fixes (better exemplars vs. graph coverage).

**Intent errors trace to one rule-ordering artifact.** Accuracy is 100% for five of eight intents; `area_summary` (33%) and `education_impact` (67%) account for nearly all errors, plus a partial miss in `road_status` (75%). All four misclassification patterns share one cause: the keyword rules fire in a fixed order, and the `earthquake_info` rule outranks `road_status` and `area_summary`, so any question merely mentioning "earthquake" is captured by it regardless of intent (e.g. "What roads are inaccessible after the earthquake?" $\rightarrow$ `earthquake_info`).

**A fifth of questions are unstable across identical runs.** 24 of 30 questions take the same route on all 3 repeats; 6 flip between `dynamic` and `fallback` on identical input, a property of the 1.5B local generator. Separately, 8 questions including all three out-of-scope ones fall back on every run, a systematic rather than stochastic gap.

### 4.5 Study 2: Answer Quality Results

Both judges independently rank Clarity highest and Reasoning Support lowest, and agree within one point 87 to 97% of the time. The system writes fluently, but connecting evidence to the conclusion is its consistent weak point.

### 4.6 Joint Analysis: Does Route Affect Quality?

Joining the two studies per question (14 dynamic, 16 fallback answers), the mean quality difference (dynamic $-$ fallback) is $-0.07$ (gpt-oss-120b) and $-0.21$ (Gemini), with inconsistent signs across criteria - Helpfulness favours dynamic, Evidence Grounding favours fallback. With only 14-16 answers per group this is underpowered; we read it as no reliable difference in perceived quality between routes.

### 4.7 Discussion

Each study alone suggests a system that works; read together, they reveal two failures. Intent errors trace to a linguistic ambiguity no keyword-based classifier can resolve, and high grounding scores cannot tell a correct answer from a fluent, well-cited answer to the wrong question.

**One word, two roles: topical vs. intentional.** The four misclassified questions all contain the word *earthquake*, and the keyword rules check for it first. Because `earthquake_info` sits above `road_status` and `area_summary` in the rule order, any question that mentions the event is routed there, even when the question is about roads or regional damage. The problem is that *earthquake* does two jobs. In *"Tell me about the earthquake"* it names

| Criterion | gpt-oss-120b | Gemini 3.1 FL |
|---|---|---|
| Helpfulness | 3.73 | 4.17 |
| Clarity | 4.70 | 4.83 |
| Evidence Grounding | 4.33 | 4.13 |
| Appropriateness | 4.03 | 4.43 |
| Reasoning Support | 3.67 | 4.13 |

Table 5: Mean LLM-judge scores on a 1-5 scale for 30 answers per judge.

the information need. In *"What roads are inaccessible after the earthquake?"* it just sets the context the user wants road information, not earthquake information. A keyword rule cannot tell these apart. Reordering the rules would not help either; it would just move the error to a different set of questions. What is needed is a classifier that can distinguish whether a word is the point of the question or part of its background.

**Grounded language is not the same as a grounded answer.** Evidence Grounding scores 4.33 and 4.13 across the two judges (Table 5), among the highest of the five criteria yet Study 1 shows that over a third of answers come from fallback queries that may answer at a different geographic level or intent than the question asked. These answers score well because the language genuinely looks grounded: they name real entities, cite real figures, and use the right geographic terms. The problem is not in how the answer is written but in what it is written *about*. A question about a parish answered with state-level totals uses correct, well-cited language throughout, it is just not the answer the user asked for. Appropriateness (4.03, 4.43), designed to catch exactly this kind of mismatch, does not reliably detect it either, because the mismatch sits between the question and the retrieval, not between the retrieval and the text. This parallels the intent finding above: in both cases, what the language looks like on the surface a keyword present, an entity correctly cited is mistaken for what the language actually means.

**Implications for designing support systems.** Both failures point to the same lesson for responsible AI in disaster management: fluent, well-grounded language by an AI agent is not evidence of a correct answer. For those who design and develop decision-support systems, evaluation cannot stop at the answer text; each answer should be read alongside the pipeline route that produced it and checked against the question that was asked.

## 5 Conclusion

When a KG-powered QA system grounds its answers in background knowledge, it is tempting to trust them. Our critical evaluation of RISE, conducted through two complementary studies on the same 30 questions, shows that this trust must be earned: standard answer-quality metrics confirm that an answer is fluent and grounded, not that it addresses the question actually asked.

The stage-wise evaluation revealed that 87% of questions are classified correctly, that every question receives an answer, and that nearly two-thirds are answered by a query the system generated rather than a pre-written fallback. The LLM-as-a-judge evaluation confirmed that the resulting answers score well on clarity, evidence grounding, and appropriateness across two independent judges. Taken separately, both studies suggest a system that works. Read together, they tell a different story.

Intent misclassification traces to a linguistic ambiguity that no keyword-based classifier can resolve: the word *earthquake* serves as context in one question and as the information need in another, and lexical matching cannot tell them apart. When a misclassified question still produces a valid query downstream, the error becomes invisible to both studies individually-Study 1 no longer flags it once the query succeeds, and Study 2 cannot detect it because the answer is genuinely grounded in the evidence returned. Separately, high scores on Evidence Grounding and Appropriateness do not distinguish a correct answer from a well-grounded answer to a different question, because the rubric scores the text against the evidence it cites rather than against the question that was asked. In both cases, what the language looks like on the surface is mistaken for what the language actually means.

For disaster management, where a confidently wrong answer can misdirect coordination decisions, these findings carry a common lesson for responsible AI and for how effective support systems are designed and developed: fluent, well-grounded language by an AI agent is not evidence of a correct answer, and evaluating these systems on their text alone is not enough. Fluent language is evidence that the system writes well, not that it answered the question.

## Limitations

Our central claim is that a well-scored answer can still answer the wrong question. We establish *why*

this happens: the judge rubric scores text against retrieved evidence rather than against the question, and Study 1 identifies which answers came from fallbacks or misrouted intents. We do not, however, establish *how often* it happens, because we did not manually check each answer against its question. That check is the natural next step, and the 30-question set makes it feasible. The same small set is also our second limitation: it is large enough to surface systematic failures such as the rule-ordering error, but too small to estimate per-intent accuracy or to detect modest quality differences between routes. Finally, these failures belong to one system built for one event, and other pipelines will fail in their own ways. What carries over is the approach: tracing each question through the pipeline and comparing that trace with an independent judgment of the final answer works for any KG- or retrieval-based QA system whose intermediate steps can be logged.

## Acknowledgments

The authors acknowledge the partial research support from the National Science Foundation grant # 2531369.

## References

Heajun An, Connor Ng, Sandesh Sharma Dulal, Junghwan Kim, and Jin-Hee Cho. 2026. Grounded but misleading: Evaluating semantic alignment in ai-generated security explanations. *arXiv preprint arXiv:2602.05056*.

Gaurav Arora, Shreya Jain, and Srujana Merugu. 2024. Intent detection in the age of llms. In *Proceedings of the 2024 conference on empirical methods in natural language processing: Industry track*, pages 1559–1570.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, et al. 2023. Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls. *Advances in Neural Information Processing Systems*, 36:42330–42357.

Jasper Linders and Jakub M Tomczak. 2025. Knowledge graph-extended retrieval augmented generation for question answering. *Applied Intelligence*, 55(17):1102.

Hemant Purohit, Rajaraman Kanagasabai, and Nikhil Deshpande. 2019. Towards next generation knowledge graphs for disaster management. In *2019 IEEE 13th international conference on semantic computing (ICSC)*, pages 474–477. IEEE.

Larissa Pusch and Tim OF Conrad. 2024. Combining llms and knowledge graphs to reduce hallucinations in question answering. *arXiv preprint arXiv:2409.04181*.

Larissa Pusch and Tim OF Conrad. 2025. Combining llms and knowledge graphs to reduce hallucinations in biomedical question answering. *BioMedInformatics*, 5(4):70.

Mengying Wang, Nicolaas Jedema, Rahul Pandey, RaviKiran Krishnan, Jens Lehmann, and Yinghui Wu. 2026. Graph query generation with constraint-guided large language agents. In *2026 IEEE 42nd International Conference on Data Engineering (ICDE)*, pages 3813–3819.

Yiming Xiao, Ankit Basu, Kai Yin, Sahil Vartak, Christian Swords, and Ali Mostafavi. 2026. Disasterlex: An expert concept-to-schema knowledge graph for geospatial reasoning in disaster analytics. *arXiv preprint arXiv:2605.30538*. Accepted to EMNLP 2026.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 3911–3921.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Yujia Zhou, Wenbo Zhang, Jingying Shao, Yan Liu, Xiaoxi Li, Jiajie Jin, Hongjin Qian, Zheng Liu, Chaozhuo Li, Jason Chen Zhang, et al. 2024. Trustworthiness in retrieval-augmented generation systems: A survey. *arXiv preprint arXiv:2409.10102*.