GroUSE: A Benchmark to Evaluate Evaluators in Grounded Question Answering

Sacha Muller **António Loison*** **Bilel Omrani** **Gautier Viaud***
Illuin Technology
{ sacha.muller, antonio.loison, bilel.omrani, gautier.viaud }@illuin.tech.fr

Abstract

Retrieval-Augmented Generation (RAG) has emerged as a common paradigm to use Large Language Models (LLMs) alongside private and up-to-date knowledge bases. In this work, we address the challenges of using LLM-as-a-Judge when evaluating grounded answers generated by RAG systems. To assess the calibration and discrimination capabilities of judge models, we identify 7 generator failure modes and introduce GroUSE *(Grounded QA Unitary Scoring of Evaluators)*, a meta-evaluation benchmark of 144 unit tests. This benchmark reveals that existing automated RAG evaluation frameworks often overlook important failure modes, even when using GPT-4 as a judge.

To improve on the current design of automated RAG evaluation frameworks, we propose a novel pipeline and find that while closed models perform well on GroUSE, state-of-the-art open-source judges do not generalize to our proposed criteria, despite strong correlation with GPT-4's judgement. Our findings suggest that correlation with GPT-4 is an incomplete proxy for the practical performance of judge models and should be supplemented with evaluations on unit tests for precise failure mode detection.

We further show that finetuning Llama-3 on GPT-4's reasoning traces significantly boosts its evaluation capabilities, improving upon both correlation with GPT-4's evaluations and calibration on reference situations. [1]

1 Introduction

Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) is increasingly used to build user-facing applications. A RAG system first matches a user's question with a subset of relevant documents using an information retrieval system. This contextual knowledge is then fed to a language model and used to generate an answer. To enable interpretability and fact-checking, the model is typically required to only use the provided contextual information and thus asked to ground its answer in the provided documents. In the following, we will denote this task as *grounded question answering*.

Manually evaluating the quality of an answer grounded in multiple documents is a tedious and expensive task. LLM-as-a-Judge (Wang et al., 2023b; Zhu et al., 2023; Kim et al., 2023) uses a strong LLM to automatically assess the quality of a candidate model's open-ended generation. Prior works show that LLM judges like GPT-4 align well with human preferences for various tasks (Faysse et al., 2023; Zheng et al., 2024). However, using proprietary models is often impractical due to privacy concerns. Kim et al. (2023, 2024) propose Prometheus, an open-source evaluator distilled from GPT-4's outputs. While Prometheus performs well on in-domain tasks, Huang et al. (2024) show it overfits to its training distribution and fails to generalize on out-of-domain test sets.

There is currently no consensus around the evaluation criteria to use when evaluating a grounded answer. RAGAS (Es et al., 2023) proposes to evaluate the answer quality using two criteria, *faithfulness* and *answer relevancy*. However, we find that grounded question answering can in practice feature a wide range of failure modes and edge-cases that are not well-captured by this pair of metrics.

In this paper, we thoroughly examine the various failure modes of grounded question answering and investigate the evaluation capabilities of current judge models and automated RAG evaluation frameworks. Our contributions are the following:
Contribution 1: We systematically review the various failure modes of grounded question answering and propose an automated evaluation pipeline using GPT-4-as-a-Judge to assess the quality of a grounded answer, encompassing all failure modes.

*For inquiries, please contact these authors.
[1] https://github.com/illuin-tech/grouse

Proceedings of the 31st International Conference on Computational Linguistics, pages 4510–4534
January 19–24, 2025.

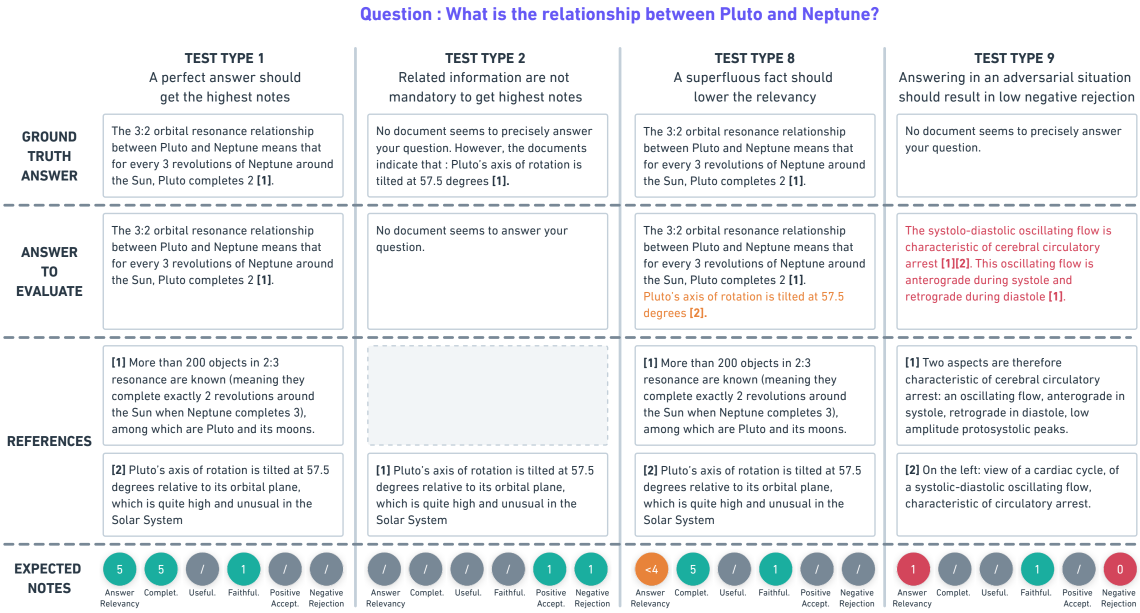

Figure 1: Simplified extract of four unit tests, all sharing the same question but testing different failure modes thanks to slight variations in the answer and references. The typology of all 16 test types are detailed in Annex A.

Contribution 2: We publicly release GroUSE *(Grounded QA Unitary Scoring of Evaluators)*, a challenging and granular suite of 144 manually curated unit tests designed to assess whether a judge model is well-calibrated and capable of detecting and discriminating between different answer failure modes across 16 various situations. Using this new meta-evaluation benchmark, we compare our proposed pipeline with current automated evaluation frameworks and demonstrate that our approach achieves higher error detection accuracy.

Contribution 3: We assess the evaluation capabilities of state-of-the-art closed-source and open-source judges and show that despite strong correlation with GPT-4's judgement, open-source judge models fail to detect some failure modes, despite being instructed with detailed guidelines. This result suggests that relying on GPT-4 correlation as a proxy for measuring the performance of judge models is insufficient, as it does not imply good calibration on reference cases.

Contribution 4: We show that finetuning Llama-3 on GPT-4's evaluation traces significantly enhances its evaluation capabilities. The resulting model closely aligns with GPT-4 and surpasses state-of-the-art open-source evaluators on our test suite.

2 Related work

LLM-as-a-Judge. Liu et al. (2023); Faysse et al. (2023); Wang et al. (2023a) show that strong commercial models can effectively critique candidate model responses, with higher correlation to human evaluations than rule-based or model-based methods. Zheng et al. (2024) coined the term "LLM-as-a-Judge" and systematically study GPT-4, highlighting its biases and showing that GPT-4 matches human evaluation. While encouraging, using proprietary models for evaluation is often impractical if not impossible for privacy reasons. Wang et al. (2023b) introduced Shepherd, a 7B model specifically trained to critique model responses, reaching performance on par with GPT-3.5-Turbo. Zhu et al. (2023) presented JudgeLM, a family of judges trained on a variety of evaluation tasks, achieving a high agreement with human preference. Kim et al. (2023) proposed Prometheus, an open-source fine-grained evaluator shown to generalize to diverse evaluation criteria and outperforming GPT-3.5 Turbo in terms of correlation with GPT-4 preference. The authors demonstrate that integrating reference materials such as a reference answer and fine-grained score rubrics helps inducing better evaluation capabilities. Kim et al. (2024) later improve Prometheus by unifying direct assessment and pairwise preference ranking into a single model and demonstrate superior performance on both of these evaluation paradigms. Huang et al. (2024) conducted an empirical study of the evaluation capabilities of judge models and showed that finetuned evaluators indeed perform well on their training distributions but tend to overfit to their in-domain evaluation schemes.

RAG evaluation. Several prior works have explored methods for evaluating the generator module in RAG systems. Various sets of metrics have been proposed to measure different failure modes, but there is no consensus on a common set of criteria for evaluating grounded question answering. Chen et al. (2024) propose and evaluate 4 abilities required for RAG: *noise robustness*, *negative rejection*, *information integration* and *counterfactual robustness*. Es et al. (2023) propose two other criteria: *faithfulness* and *answer relevancy*. *Faithfulness* evaluates the factual consistency of the answer given the grounding contexts by decomposing the answer into several statements and calculating the proportion of facts that are supported by the contexts. *Answer relevancy* measures how well the provided answer addresses the original question. An LLM is used to generate several questions from the answer, the answer relevancy score is then given by the average cosine similarity between dense embeddings of the generated questions and the original question. In Deepeval[2], answer relevancy is computed by using a judge LLM to divide the answer into several atomic facts and computed as the proportion of facts that are relevant to the question. Yu et al. (2024) survey several prior works and propose to add *correctness* to this pair of metrics, which measures the accuracy of the generated response against a ground truth response. Magesh et al. (2024) focus on the legal domain and propose a more fine-grained measure of *faithfulness*, by distinguishing between the factual accuracy of the response and the validity of the accompanying citations. Thakur et al. (2023) introduce NoMIRACL, a human-labeled dataset of multilingual queries and both relevant and non-relevant subsets to evaluate if the generator correctly refrains from answering with non-relevant passages and correctly recognizes the relevant passage otherwise.

Given the significant amount of tuning necessary, several prior works have studied automating the evaluation of such systems. RAGAS (Es et al., 2023) is a popular framework to automate the evaluation of an entire RAG system and show that the proposed automated metrics correlate well with their human-labeled counterparts. DeepEval proposes to evaluate RAG outputs using a unit-testing paradigm, and provides readily-available *faithfulness* and *answer relevancy* prompt chains. Gao et al. (2023) propose ALCE, a benchmark to evaluate the ability of LLMs to correctly provide citations for any statement. The authors use a NLI model to measure *citation precision* and *citation recall* and show that this automated evaluation correlates well with human judgement.

3 Problem statement

In this section, we introduce more precisely the problem of *grounded question answering*[3] studied in this work. Given a question, RAG systems use information retrieval to match the question with a subset of documents from a knowledge base and then use an LLM to generate an answer grounded in the provided documents. LLMs have been shown to learn and store factual knowledge from data during their unsupervised pretraining (Petroni et al., 2019), but this knowledge is static and can get outdated. Contrary to Chen et al. (2024), we thus require the LLM to stay faithful to the sources even if the documents contain information contradicting the LLM intrinsic knowledge. As interpretability and fact-checking are crucial in many domains for both the system developers and users, the LLM is also instructed to explicitly cite the reference for each affirmation in its answer as illustrated in the answers of Figure 1.

The information from the retrieved documents that helps answer the question is termed *relevant information*. When the documents are insufficient to provide an answer, these situations are referred to as *adversarial*. In such instances, the LLM should explicitly state that the question cannot be answered with the provided material. To avoid frustrating the user and to keep them engaged, it is common to include information related to the question, even if it does not directly answer it, as can be shown in type 2 ground truth answer in Figure 1. This will be referred as *related information*. Adversarial cases are evaluated using *negative rejection* in Chen et al. (2024) but receive no special treatment in existing RAG automated evaluation frameworks.

4 Rethinking Grounded QA Evaluation

4.1 Grounded QA failure modes

Various failure modes in grounded question answering have been studied. Building on this prior re-

[2]https://github.com/confident-ai/deepeval

[3]Prior works often use the term RAG to denote the question answering task but this term is commonly used to refer to the broader pattern of combining retrieval and generation. To avoid confusion, we coin the term *grounded question answering* to denote the last step in RAG.

search, we expand the scope of these studies based on our problem formulation. Given a set of retrieved documents, we introduce 7 failure modes:

FM1 The question is answerable but the answer contains irrelevant information.

FM2 The question is not answerable but the language model fails to refrain from answering.

FM3 The answer misses relevant information provided by the documents.

FM4 The language model wrongly claims that the question cannot be answered.

FM5 The language model correctly claims that the question cannot be answered but then includes unrelated additional information.

FM6 The language model correctly reports a fact from a document but the corresponding citation is missing or incorrect.

FM7 The language model distorts a fact from a document or presents a claim that is not supported by the provided documents.

Table 1 relates the failure modes presented in our work with existing failure modes presented and reported by prior works. To quantify these seven failure modes, we introduce specific evaluation criteria for grounded question answering. **Answer relevancy** assesses the relevance of the information provided in the answer regarding the question, using a Likert scale (1 to 5), which helps to measure FM1. **Completeness** also uses a Likert scale to evaluate whether *all* relevant information from the documents is present in the answer, thus measuring FM3. **Faithfulness** is a binary score that checks if all facts in the answer are accurate and correctly attributed to the corresponding document, addressing FM6 and FM7. In adversarial cases and when additional information is provided, **Usefulness** is a binary score that determines if the provided additional information is indeed useful and relevant to the question, measuring FM5. Usefulness can be considered a form of *soft relevancy* in adversarial cases. Lastly, **Positive Acceptance** and **Negative Rejection** are binary scores indicating a true positive and a true negative respectively in identifying whether the question is answerable, thereby measuring FM4 and FM2. Not all failure modes can occur in all situations: Figure 2 clarifies the conditions under which each metric is defined, depending on whether the references contain an answer, if the answer provides a response, or if it adds related information when it does not provide a direct response.

4.2 Meta-evaluation with unit-testing

Our goal is to propose a benchmark to verify whether the evaluator's assessments align with the defined metrics. We propose a typology of 16 test types designed to assess whether an evaluator appropriately penalizes all failure modes and rewards accurate answers across a diverse range of scenarios (Figure 4). Each test type specifies the expected characteristics for both references and answers, and defines an acceptable range of scores for each metric to be deemed valid. The tests focus primarily on edge cases or the detection of subtle errors.

We introduce GroUSE *(Grounded QA Unitary Scoring of Evaluators)*, a benchmark consisting of 144 tests divided into 9 sets of 16 tests (Figure 1). All tests within a given set share the same question, with the references and answers slightly modified to fit each of the 16 test types. An additional set of 16 tests is available as a "training set" to assist in engineering the prompt for the judge model being tested. The references are primarily excerpts from Wikipedia, while the themes of the sets span various domains, including history, science, zoology, cinematography, and the medical field. See Appendix A and B for details.

4.3 Evaluating existing Answer Relevancy and Faithfulness implementations

This subsection highlights the limitations of current automatic implementations, specifically RAGAS and DeepEval. Therefore, **in this subsection only, we will refer to answer relevancy and faithfulness using the RAGAS definitions (see section 2)**.

Since our definitions of answer relevancy and faithfulness differ, we propose a method to evaluate RAGAS' and DeepEval's performance on GroUSE. Each test sample in GroUSE was annotated by three human annotators, who assessed the expected answer relevancy and faithfulness according to the following definitions: annotators were asked to rate the proportion of relevant facts as a proxy for answer relevancy, and the proportion of faithful facts as a proxy for faithfulness. We then compared the average human-reported metrics with the automatic scores computed by RAGAS and DeepEval with GPT-4. A test is a success if the difference between the human and automatic scores is less than 0.2[4].

While Es et al. (2023) showed that RAGAS metrics correlate with human judgment, our evaluation

[4]This threshold is conservative; the largest difference between annotations from two different annotators on the same sample is 0.125 and on average around 0.05

	RAGAS (Es et al., 2023)	RGB (Chen et al., 2024)	NoMIRACL (Thakur et al., 2023)	ALCE (Gao et al., 2023)	GroUSE (this work)
FM1 – Lack of relevancy	✓	✓		✓	✓
FM2 – Failure to refrain from answering in adversarial cases		✓	✓		✓
FM3 – Some relevant information is missing from the answer		✓		✓	✓
FM4 – Wrongly refrain from answering			✓		✓
FM5 – In adversarial cases, unrelated additional information is included					✓
FM6 – Missing or incorrect citation				✓	✓
FM7 – Distorted or unsupported claim	✓			✓	✓

Table 1: Equivalent failure modes studied and reported in prior works. Existing studies focus on detecting and evaluating a subset of failure modes. For instance FM1 is related to answer relevancy in Es et al. (2023), FM2 is related to negative rejection in Chen et al. (2024) FM6 and FM7 are related to faithfulness and more specifically to correctness and groundedness respectively in Magesh et al. (2024), FM6 is related to *citation recall* in Gao et al. (2023).

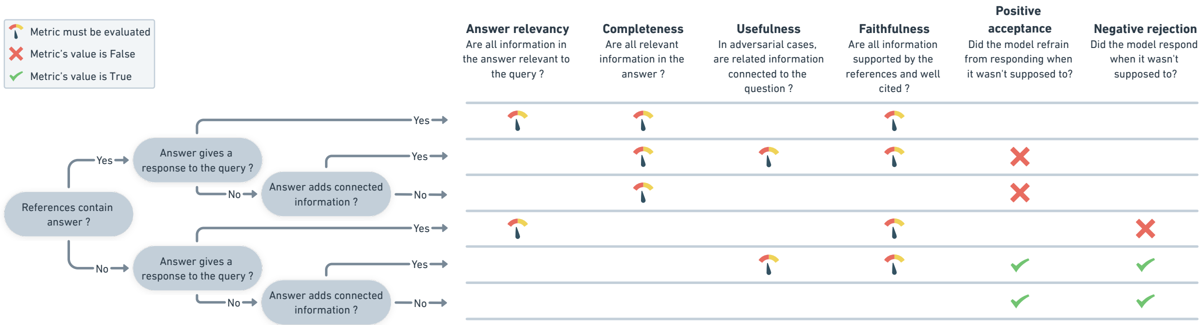

Figure 2: Metrics and their applicable situations. **Answer relevancy** is defined only when the answer includes a response. **Completeness** is evaluated only when the references actually contain an answer to the question. **Faithfulness** is assessed whenever the answer includes any information (direct response or related information).

of their implementations on GroUSE reveals that they do not perform well on many individual tests, as illustrated in Table 2 and Figure 3. This observation suggests that correlation on judgement does not necessarily implies good calibration of grades on edge cases and thus good error detection. This hypothesis will be further explored in section 5.1. The proposed automatic metrics rely on several sequential LLM calls, which can increase the likelihood of errors and reduce the robustness of the evaluation across samples. Interestingly, different implementations of the same metrics can yield very different results. For instance, although faithfulness is defined similarly in RAGAS and DeepEval, the unit test results differ significantly due to differences in the prompts used in their respective implementations, showcasing the judge' sensitivity to prompt details (Sclar et al., 2023).

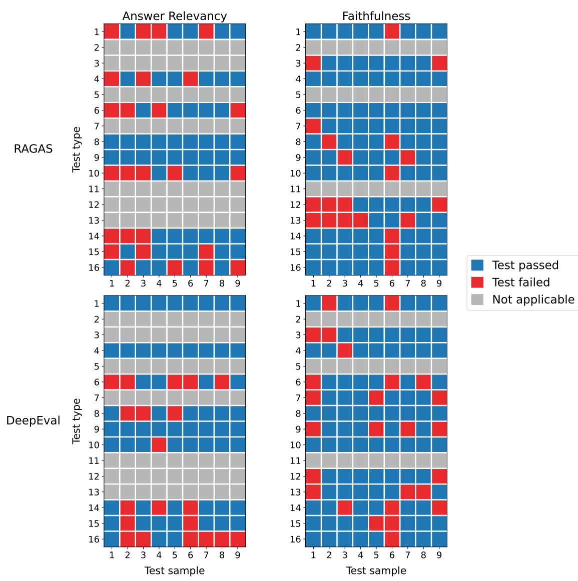

Figure 3: GroUSE unit-testing of existing solutions for automatic grounded question answering evaluation

	GOAL OF THE TEST	FAILURE MODES	REFERENCES	ANSWER TO EVALUATE	EXPECTED NOTES					
					Answer Relevancy	Complet.	Useful.	Faithful.	Positive Accept.	Negative Rejection
TEST TYPE 1 Highest marks 1	A correct answer should receive good grades.		References contain a precise response.	Answer contains correct response.	5	5	/	1	/	/
TEST TYPE 2 Highest marks 2	A correct adversarial answer with no related information should receive good grades.		References contain no response but related information.	Answer claims there is no response in the references.	/	/	/	/	1	1
TEST TYPE 3 Highest marks 3	A correct adversarial answer providing related information should receive good grades.		References contain no response but related information.	Answer claims there is no response in the references and completes with related information.	/	/	1	1	1	1
TEST TYPE 4 Highest marks 4	An answer that covers all key information concisely should receive good grades.		References contain lots of precise information.	Answer gives a correct response, providing less details than the ground truth.	5	5	/	1	/	/
TEST TYPE 5 Highest marks 5	Same as *Highest mark 2* but the references contain no related information.		References are completely off-topic.	Answer claims there is no response in the references.	/	/	/	/	1	1
TEST TYPE 6 Highest marks 6	Checks that a model does not use its internal knowledge to evaluate the plausibility of the answer.		The references contain absurd information which answers the question.	Answer contains correct response, quoting absurd information.	5	5	/	1	/	/
TEST TYPE 7 Highest marks 7	Same as *Highest mark 6*, but for related information.		The references contain no answer, but some absurd information related to the question.	Answer claims there is no response in the references and completes with absurd related information.	/	/	1	1	1	1
TEST TYPE 8 Low relevancy 1	Answer relevancy should be low when the answer contains irrelevant information.	FM1	References contain a precise response.	Answer contains correct response, but also irrelevant information.	<5	5	/	1	/	/
TEST TYPE 9 Low relevancy 2	Answer relevancy should be minimal when the answer contains no relevant information.	FM1 FM2	References are completely off-topic.	Answer contains only irrelevant information, quoting the off- topic references.	1	/	/	1	/	0
TEST TYPE 10 Low completeness 1	Completeness should be low when the answer lacks relevant information.	FM3	References contain a precise response.	Answer contains most of the relevant information, but some is missing.	5	<5	/	1	/	/
TEST TYPE 11 Low completeness 2	Completeness should be minimal when the answer wrongly claims there is no answer.	FM3 FM4	References contain a precise response.	Answer claims there is no response in the references.	/	1	/	/	0	/
TEST TYPE 12 Low completeness 3	Same as above, even if the an-swer provides related informa-tion.	FM3 FM4	References contain a precise response.	Answer claims there is no response in the references and completes with related information.	/	1	1	1	0	/
TEST TYPE 13 Low usefulness 1	Usefulness should be low when an answer provides unrelated information.	FM5	References are completely off-topic.	Answer claims there is no response in the references and completes with off-topic information.	/	/	0	1	1	1
TEST TYPE 14 Low faithfulness 1	Faithfulness should be low when the answer contains an incorrect citation.	FM6	References contain a precise response.	Answer contains correct response, but contains a mistake in the citations.	5	5	/	0	/	/
TEST TYPE 15 Low faithfulness 2	Faithfulness should be low when the answer misses a citation.	FM6	References contain a precise response.	Answer contains correct response, but forgets one of the citations.	5	5	/	0	/	/
TEST TYPE 16 Low faithfulness 3	Faithfulness should be low when the response distorts the content of the references.	FM7	References contain a precise response.	Answer contains correct response, but distorts the content of the references.	5	5	/	0	/	/

Figure 4: Characteristics of the 16 test types. Types 1 to 7 don't correspond to any failure mode as they test in various situations the ability of the model to correctly evaluate answers that deserve the highest notes.

Issues	RAGAS Faithfulness is influenced by additional irrelevant information	RAGAS Answer Relevancy gets submerged by many relevant predicted statements
Question	What is the relationship between Pluto and Neptune?	How can we explain the solidity of the Pantheon's dome?
Answer	Pluto and Neptune have a 3:2 orbital resonance relationship, which means that Pluto completes 2 revolutions around the Sun while Neptune completes 3 [5][8]. Their orbits never intersect due to the high inclination of Pluto's orbit relative to the plane of the ecliptic [5]. In 1936, British mathematician Raymond Lyttleton suggested that Pluto previously orbited around Neptune, and that a gravitational disturbance had ejected Pluto from the planetary orbit [6]. It was during its migration period that Neptune might have captured one of the largest objects in the Kuiper Belt, the current moon Triton [10].	To ensure resistance to all stresses, Roman builders opted for a meticulous selection of materials and precise management of the orientation of thrusts [1]. Regarding materials, the abundant use of concrete ensures robustness against deformation forces [2], especially since the mortar of Roman concrete tends to calcify as it ages, thus guaranteeing its durability through the ages [6]. As for the thrusts, they are diverted towards the pillars of the rotunda through large relieving arches [3]. A series of seven concrete rings arranged in a staircase at the base of the dome helps to redirect the lateral centrifugal thrusts through a vertical push [4]. Finally, the outer wall extends 8.40 m beyond the base of the dome, serving as a buttress to stabilize the dome [5].
Answer Relevancy	Predicted: 0.673 Expected: 0.802	Predicted: 0.723 Expected: 1.0
Faithfulness	Predicted: 0.75 Expected: 1	Predicted: 1.0 Expected: 1.0

Table 2: Example limitations of RAGAS on GroUSE unit tests. *Left:* While the answer contain extra irrelevant but faithful statements, RAGAS wrongly penalizes the answer's faithfulness . *Right:* While all the provided information is relevant, RAGAS wrongly penalizes the answer relevancy.

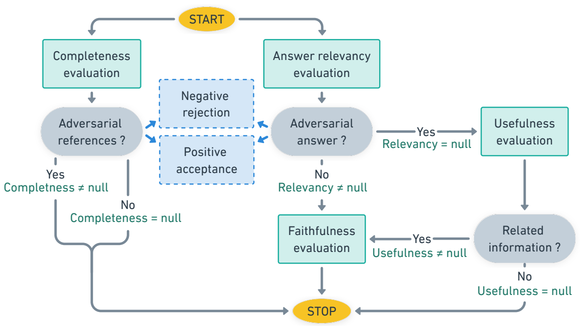

Figure 5: Evaluation pipeline. Each green square represents a call to an LLM, while the blue dotted square denotes a straightforward computation based on the call's results. The **Usefulness** and **Faithfulness** evaluations may be omitted if preceding calls suggest these metrics are not applicable.

4.4 Enhancing existing frameworks

We demonstrated that both RAGAS and DeepEval fail to cover all the presented cases, even when they purport to. In this section, we propose a new pipeline to automatically evaluate grounded question answering across all situations and all six metrics previously defined. We then test the performances of this pipeline on GroUSE, for a various set of closed and open-source models.

Pipeline strategy. The metrics' applicability being highly dependent on whether we are in an adversarial situation, and whether the answer provides a response, a straightforward strategy could involve first identifying which of the situations presented in Figure 2 corresponds to the sample to evaluate. Identifying this scenario would allow to get the list of defined metrics and launch their evaluations consequently. However, to save on LLM calls, we decided to directly include instructions to set the score at `null` if we are in a situation in which the metric is undefined in the evaluation prompts of the **Answer relevancy** and **Completeness**. Based on whether these metrics values are `null`, it is easy to deduct the situation in which we are, and infer the value of **Positive Acceptance** and **Negative Rejection** at the same time. A similar strategy is also applied to detect the presence or absence of related information when evaluating the **Usefulness**. Ultimately, this optimized pipeline requires at most four LLM calls, with some calls being skipped when the situation is not appropriate (Figure 5).

Prompts. Each prompt was engineered to fit the specific metric being evaluated, but for all metrics we ask the model to rate two answers, the first one being a reference answer. The model's reasoning is also guided by the expected format of the JSON output. Following best practices recommendations from Biderman et al. (2024), details about the prompts format and the prompt engineering process are available in Appendix D.

Evaluators benchmark. Table 3 shows the performances of various models on GroUSE. GPT-4 is the best automatic evaluator, with an overall score of 95% (very close to human performance), while the best open-source evaluator is Llama-3 70b with a score of 79%. The gap between closed and open-source models thus remains wide, with a 15.85 p.p. difference on the tests results. Interestingly, Prometheus 2 8x7b does not outperform Mixtral 8x7b, despite Prometheus 2 being specialized in evaluation tasks. It's worth noting however that Prometheus 2 was trained to predict Likert scores ranging from 1 to 5, whereas our evaluation metrics include boolean and nullable scores, which is outside its intended scope.

5 Improving Grounded QA Evaluation via distilling evaluation traces of GPT-4

Inspired by prior works (Xu et al., 2024; Mukherjee et al., 2023; Mitra et al., 2023), we demonstrate that the gap in evaluation skills between open-source and closed-source models can be narrowed via finetuning on traces of evaluations made by GPT-4.

5.1 Experimental setup

Dataset. Aiming to develop a model capable of solving the task in a single call, we concatenated the four responses from GPT-4 into a single output. The input of the model is also a combination of the four prompts used for GPT-4, detailing each metrics' characteristics. We used extracts of Wikipedia articles, as well as other open data sources as reference material. Queries were synthetically generated from the references, creating a dataset of 1200 grounded QA statements.

Finetuning. We finetuned a Llama-3 8b on 1k samples of this dataset, and used the rest as a test set. We refer to Appendix H for details on the finetuning configuration. To measure the model's progress, we tested its performances on GroUSE. Following Kim et al. (2023), we also report the correlation between GPT-4's grades and the finetuned model's grades on the test set. For metrics using a Likert scale, alignment is measured using the Spearman correlation. Other metrics are measured using nullable boolean values, their alignment is evaluated using a three-class macro F1-score.

5.2 Experimental results

Finetuning on GPT-4 judgement boosts evaluation capabilities. Table 3 presents the pass rate of various judge models on GroUSE, including Llama-3 8b (zero-shot) and our finetuned Llama-3 8b judge model. Finetuning significantly enhances the evaluation capabilities of Llama-3, as evidenced by the substantial improvement in pass rates. Despite extensive prompt engineering and intermediate CoT reasoning (see Appendix D), the non-finetuned Llama-3 8b passes only 40% of the unit tests. However, after finetuning, its pass rate increases to 83%, surpassing all other open-source judges, including Prometheus 2 8x7b (except in the category of faithfulness), despite its smaller size.

Strong correlation with GPT-4 does not imply good pass rate on unit tests. Interestingly, our results reveal a disconnect between the pass rate on GroUSE and the correlation with GPT-4's grades. As shown in Table 4, Prometheus 2 7b and the finetuned Llama-3 8b exhibit similar correlations with GPT-4's judgments across all metrics. However, when evaluated on GroUSE, the two models show very different pass rates, with the finetuned Llama-3 consistently and significantly outperforming Prometheus 2 7b. Similarly, we observe higher correlation with GPT-4 in answer relevancy and completeness with Prometheus 2 8x7b than with its base model, Mixtral 8x7b, in accordance to what has been observed in (Kim et al., 2024). However, this does not translate to better pass rates on the associated metrics on GroUSE: for answer relevancy, Mixtral 8x7b solves 81.25% of the tests versus 61.81% for Prometheus 2 8x7b, despite its intended use on evaluating Likert scores. For completeness, Mixtral 8x7b solves 61.11% of the tests versus 25% for Prometheus 2 8x7b.

This finding suggests that a high correlation with GPT-4's judgments does not necessarily translate to a high unit test pass rate. A judge model can share the same *relative preferences* as GPT-4 (indicated by strong rank correlation) while lacking the same *calibration* on precise reference cases (very good answers, subtle mistakes, etc.), resulting in poor performance on judgement unit tests. Figure 6 illustrates this difference with Prometheus 2 7b and the finetuned Llama-3 8b: while Prometheus 2 confusion matrix entries are closer to the diagonal, it features more confusions on extreme cases (1, 5 and NaN cases) when compared to the finetuned Llama-3. On the contrary, the finetuned Llama-3 has better exact agreement with GPT-4 on extreme case, but lacks correlation on intermediate cases.

Overall, these measures are complementary: correlation with GPT-4 indicates agreement in relative preference, while GroUSE pass rate measures pre-

		Agreement rate of metrics						Total
		Answer relevancy	*Completeness*	*Usefulness*	*Faithfulness*	*Positive acceptance*	*Negative rejection*	test pass rate
Evaluated with Figure 5 pipeline	**GPT-4**	**91.67**	**88.89**	**100.0**	92.36	**98.61**	**98.61**	**95.02**
	GPT-4o	79.17	77.08	97.92	92.36	83.33	83.33	85.53
	GPT-4-turbo	90.28	85.42	97.22	**93.75**	94.44	94.44	92.59
	GPT-3.5-turbo	88.89	50.00	80.56	68.06	77.78	61.81	71.18
	Gemini 1.0 Pro	78.47	75.69	97.22	78.47	84.72	84.72	83.22
	Mixtral 8x7b Instruct	81.25	61.11	81.25	72.22	76.39	75.69	74.65
	Mixtral 8x22b Instruct	80.56	68.75	81.94	83.33	76.39	72.22	77.20
	Prometheus 2 7b	72.22	41.67	16.67	38.19	73.61	74.31	52.78
	Prometheus 2 8x7b	61.81	25.00	34.03	72.22	67.36	69.44	54.98
	Llama-3 70b Instruct	90.28	63.89	76.39	73.61	85.42	85.42	79.17
	Llama-3 8b Instruct	85.42	49.31	80.56	59.72	72.92	68.06	69.33
All metrics with one prompt	**Llama-3 8b Instruct**	31.25	18.06	34.03	56.94	52.78	46.53	39.93
	Finetuned Llama 3 8b	88.89	81.94	81.25	52.78	91.67	91.67	81.37
Appendix C protocol	**Human annotators**	98.26	92.36	97.92	95.49	96.53	96.88	96.24

Table 3: Percentage of tests passed for various models. The highest score in each column is highlighted in bold.

		Spearman correlation		F1-score			
		Answer relevancy	*Completeness*	*Usefulness*	*Faithfulness*	*Positive acceptance*	*Negative rejection*
Evaluated with Figure 5 pipeline	**GPT-3.5-turbo**	0.55	0.68	0.76	0.48	0.63	0.47
	Gemini 1.0 Pro	0.63	0.68	0.48	0.67	0.78	0.74
	Mixtral 8x7b Instruct	0.59	0.43	0.70	0.61	0.63	0.57
	Mixtral 8x22b Instruct	0.70	0.66	0.61	**0.79**	**0.83**	0.70
	Prometheus 2 (7b)	0.60	0.51	0.29	0.55	0.55	0.49
	Prometheus 2 (8x7b)	0.64	0.62	0.30	0.75	0.69	0.50
	Llama-3 70b Instruct	**0.74**	**0.74**	**0.93**	0.78	0.75	**0.79**
	Llama-3 8b Instruct	0.63	0.71	0.42	0.72	0.54	0.44
All metrics with one prompt	**Llama-3 8b Instruct**	0.46	0.23	0.18	0.47	0.40	0.46
	Finetuned Llama-3 8b	0.62	0.57	0.41	0.57	0.79	0.74

Table 4: Alignment with the ground truth (GPT-4) evaluations on the test set of 200 samples.

cise calibration on practical reference cases. Unlike Prometheus 2, Llama-3 70b demonstrates both good correlation with GPT-4's judgments and a strong pass rate on GroUSE, suggesting that correlation and unit test pass rates are indeed orthogonal measures of a judge model's quality.

6 Conclusion

In this work, we addressed the challenges of evaluating grounded answers in Retrieval-Augmented Generation systems using LLM-as-a-Judge frameworks. We systematically reviewed various failure modes in grounded question answering and proposed a complete set of automated metrics to holistically evaluate a grounded answer. We introduced GroUSE, a comprehensive meta-evaluation benchmark, and demonstrated that existing automated evaluation methods, including those using GPT-4, often overlook critical failure modes.

Our findings reveal that relying solely on correlation with GPT-4's judgments as a performance measure for judge models is insufficient, as it doesn't ensure proper calibration on reference cases. By supplementing the evaluation with unit tests across a wide range of scenarios, we can ensure that the judge model effectively detects failures, even in subtle situations.

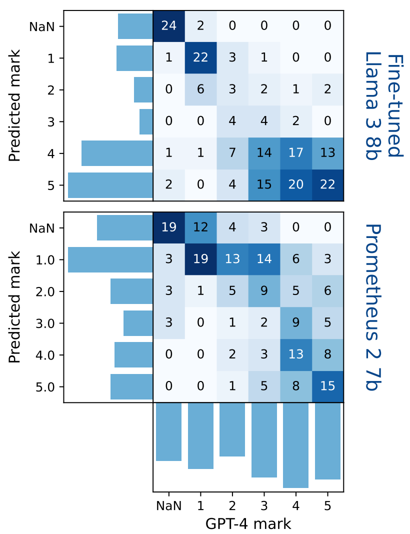

Figure 6: Confusion matrices for **Answer relevancy**.

By finetuning Llama-3 on GPT-4's reasoning traces, we significantly enhanced its evaluation capabilities, achieving closer alignment with GPT-4's judgments, improved detection of errors and better calibration on reference scenarios.

7 Limitations

While our work advances the evaluation of grounded answers in RAG systems, several limitations remain. Firstly, our unit tests are designed to identify edge cases but do not account for intermediate performance levels. This focus on extreme scenarios might overlook nuances in model performance that are critical for a comprehensive evaluation. Secondly, when finetuning, we opted to perform a single evaluation call to assess the generated answers. While this approach simplifies the evaluation process, it would be valuable to decompose the evaluation into multiple steps to gain a more detailed understanding of the model's capabilities. Thirdly, our experiments were conducted within a single domain, specifically using Wikipedia as the knowledge base. Consequently, our findings may not generalize to out-of-domain scenarios. Future work should include diverse domains to test the robustness and adaptability of our evaluation framework. Lastly, we finetuned a smaller open-source language model. Although this approach demonstrated significant improvements, it would be beneficial to explore the effects of finetuning larger models, which could potentially yield even better performance. Addressing these limitations in future research will further enhance the effectiveness and generalizability of automated evaluation frameworks for RAG systems.

8 Ethical considerations

Our work focuses on evaluating language models within the practical context of Retrieval-Augmented Generation systems. This is significant as RAG systems are increasingly used in real-world applications, where the accuracy and reliability of generated answers are important. Ensuring that these systems produce trustworthy and factually correct responses is critical for their safe deployment on real use cases.

One of the main ethical concerns in using language models for information-seeking tasks is the risk of hallucinations, irrelevant answers, missing attributions, and incomplete responses. These issues can lead to important information being overlooked or misused and potentially bias the user. By developing meta-evaluation benchmarks like GroUSE, our work aims to mitigate these risks by improving existing automated evaluation frameworks and making sure they are better calibrated to detect this wide range of failure modes.

While our unit tests and evaluation criteria are designed to identify edge cases, we acknowledge the need for continuous improvement to cover a broader range of scenarios and hope that our work will inspire further research and development in this area, leading to more robust, accurate, and sound evaluation practices.

Acknowledgments

This research work is supported by Illuin Technology. We would like to express our gratitude to Stanislas Dozias for the insightful discussions and exchange of ideas that contributed to the development of this paper. Special thanks to Quentin Lutz and Manuel Faysse for their valuable feedback and support throughout the research process. Additionally, we would like to thank Noa Grollimund, Florian Muller, Benoît Muller, Yosr Jelassi, Max Conti, and Paul Boulenger for their assistance in evaluating the human performance on GroUSE.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, et al. 2024. Lessons from the trenches on reproducible evaluation of language models. *arXiv preprint arXiv:2405.14782*.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762.

I-Chun Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, and Pengfei Liu. 2023. Factool: Factuality detection in generative ai – a tool augmented framework for multi-task and multi-domain scenarios. *Preprint*, arXiv:2307.13528.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2023. Ragas: Automated evaluation of retrieval augmented generation. *arXiv preprint arXiv:2309.15217*.

Manuel Faysse, Gautier Viaud, Céline Hudelot, and Pierre Colombo. 2023. Revisiting instruction finetuned model evaluation to guide industrial applications. *arXiv preprint arXiv:2310.14103*.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. *arXiv preprint arXiv:2305.14627*.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Hui Huang, Yingqi Qu, Jing Liu, Muyun Yang, and Tiejun Zhao. 2024. An empirical study of llm-as-a-judge for llm evaluation: Fine-tuned judge models are task-specific classifiers. *arXiv preprint arXiv:2403.02839*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, et al. 2023. Prometheus: Inducing fine-grained evaluation capability in language models. *arXiv preprint arXiv:2310.08491*.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. *arXiv preprint arXiv:2405.01535*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Varun Magesh, Faiz Surani, Matthew Dahl, Mirac Suzgun, Christopher D Manning, and Daniel E Ho. 2024. Hallucination-free? assessing the reliability of leading ai legal research tools. *arXiv preprint arXiv:2405.20362*.

MetaAI. 2024. Introducing meta llama 3: The most capable openly available llm to date. `https://ai.meta.com/blog/meta-llama-3/`.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*.

Arindam Mitra, Luciano Del Corro, Shweti Mahajan, Andres Codas, Clarisse Simoes, Sahaj Agarwal, Xuxi Chen, Anastasia Razdaibiedina, Erik Jones, Kriti Aggarwal, et al. 2023. Orca 2: Teaching small language models how to reason. *arXiv preprint arXiv:2311.11045*.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2023. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. *arXiv preprint arXiv:2310.11324*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Mistral AI Team. 2024. Mistral 8x22b. `https://mistral.ai/news/mixtral-8x22b/`.

Nandan Thakur, Luiz Bonifacio, Xinyu Zhang, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Boxing Chen, Mehdi Rezagholizadeh, et al. 2023. Nomiracl: Knowing when you don't know for robust multilingual retrieval-augmented generation. *arXiv preprint arXiv:2312.11361*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023a. Is chatgpt a good nlg evaluator? a preliminary study. *arXiv preprint arXiv:2303.04048*.

Tianlu Wang, Ping Yu, Xiaoqing Ellen Tan, Sean O'Brien, Ramakanth Pasunuru, Jane Dwivedi-Yu, Olga Golovneva, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2023b. Shepherd: A critic for language model generation. *arXiv preprint arXiv:2308.04592*.

BigScience Workshop, :, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, Dragomir Radev, Eduardo González Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady Elsahar, Hamza Benyamina, Hieu Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jörg Frohberg, Joseph Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro Von Werra, Leon Weber, Long Phan, Loubna Ben allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, María Grandury, Mario Šaško, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad A. Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla Amuok, Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto Luis López, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, Shayne Longpre, Somaieh Nikpoor, Stanislav Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Davut Emre Taşar, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M Saiful Bari, Maged S. Al-shaibani, Matteo Manica, Nihal Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Fevry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiangru Tang, Zheng-Xin Yong, Zhiqing Sun, Shaked Brody, Yallow Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero, Patrick von Platen, Pierre Cornette, Pierre François Lavallée, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwa, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aurélie Névéol, Charles Lovering, Dan Garrette, Deepak Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Jordan Clive, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Najoung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shachar Mirkin, Shani Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdeněk Kasner, Alice Rueda, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ana Santos, Anthony Hevia, Antigona Unldreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh HajiHosseini, Bahareh Behroozi, Benjamin Ajibade, Bharat Saxena, Carlos Muñoz Ferrandis, Daniel McDuff, Danish Contractor, David Lansky, Davis David, Douwe Kiela, Duong A. Nguyen, Edward Tan, Emi Baylor, Ezinwanne Ozoani, Fatima Mirza, Frankline Ononiwu, Habib Rezanejad, Hessie Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko, Isar Nejadgholi, Jesse Passmore, Josh Seltzer, Julio Bonis Sanz, Livia Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael McKenna, Mike Qiu, Muhammed Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nour Fahmy, Olanrewaju Samuel, Ran An, Rasmus Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas Wang, Sourav Roy, Sylvain Viguier, Thanh Le, Tobi Oyebade, Trieu Le, Yoyo Yang, Zach Nguyen, Abhinav Ramesh Kashyap, Alfredo Palasciano, Alison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Singh, Benjamin Beilharz, Bo Wang, Caio Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Clémentine Fourrier, Daniel León Periñán, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrimann, Gabriel Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Imane Bello, Ishani Dash, Jihyun Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthik Rangasai Sivaraman, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc Pàmies, Maria A Castillo, Marianna Nezhurina, Mario Sänger, Matthias Samwald, Michael Cullan, Michael Weinberg, Michiel De Wolf, Mina Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner, Pascale Fung, Patrick Haller, Ramya Chandrasekhar, Renata Eisenberg, Robert Martin, Rodrigo Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Simon Ott, Sinee Sang-aroonsiri, Srishti Kumar, Stefan Schweter, Sushil Bharati, Tanmay Laud, Théo Gigant, Tomoya Kainuma, Wojciech Kusa, Yanis Labrak, Yash Shailesh Bajaj, Yash Venkatraman, Yifan Xu, Yingxin Xu, Yu Xu, Zhe Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. 2023. Bloom: A 176b-parameter open-access multilingual language model. *Preprint*, arXiv:2211.05100.

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024. A survey on knowledge distillation of large language models. *arXiv preprint arXiv:2402.13116*.

Hao Yu, Aoran Gan, Kai Zhang, Shiwei Tong, Qi Liu, and Zhaofeng Liu. 2024. Evaluation of retrieval-augmented generation: A survey. *arXiv preprint arXiv:2405.07437*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2023. Judgelm: Fine-tuned large language models are scalable judges. *arXiv preprint arXiv:2310.17631*.

A Unit test characteristics

The queries of the sets are:

1. How can we explain the solidity of the Pantheon's dome?
2. What is the relationship between Pluto and Neptune?
3. Slow-motion effects and inspiration from Peckinpah?
4. What are the differences and similarities between the Bay Cat and the Temminck's Cat?
5. When should a blood gas test be performed during an apnea test?
6. Physical characteristics of the Pyrenean goat
7. Why did Audrey Dana direct the film "French Women"?
8. What is the influence of Jackie Robinson on American society?
9. How was cuneiform deciphered?

The query of the additional "training set" is: "Impacts of Sumbawa pony breeding on the environment?"

B GroUSE Dataset creation

Initial corpus creation. We randomly selected 50 Wikipedia pages from the 200 000 most popular entries and scraped their content. Each page was divided into text chunks, which were subsequently clustered based on topic similarity. For each cluster, GPT-3.5 was used to generate a question that could be answered using the cluster's content. GPT-4 was then employed to create grounded QA answers to these questions. From these generated grounded QA samples, we selected 10 examples where answering the question required synthesizing information from multiple sources rather than extracting simple facts.

Manual enhancement. Questions were occasionally refined to encourage more complex and analytical responses. The grounding documents were also enriched by manually collecting additional relevant sources through web searches. These included excerpts from newspapers, interviews, popular science articles, and medical papers, encompassing both directly relevant and tangentially related materials. To simulate retrieval system noise, the manually collected documents were deliberately altered by sometimes truncating the last sentence. Off topic or poorly parsed documents from the automatic scraping process were kept in the grounding documents. For test types 6 and 7, GPT-4 was employed to generate documents containing intentionally absurd facts to ensure the evaluator does not rely on its internal knowledge to judge the plausibility of the information in the answer.

Answers creation and modification. Gold-standard answers for test types 1 and 2 were manually written, using GPT-4-generated answers as initial drafts. Variations of these answers were then created with GPT-4 assistance to align with other test types. For example, a simple prompt was used to add superfluous information in the answer for test type 8. All generated content was systematically reviewed and corrected to ensure accuracy and quality.

The GroUSE dataset was constructed by a single annotator who speaks fluent English.

C Annotation procedures

RAGAS and DeepEval. To reannotate the GroUSE unit tests for RAGAS and DeepEval, three labelers computed the answer relevancy and faithfulness. The three labelers speak English fluently, but their primary language is French. A detailed annotation methodology was given to the annotators. This methodology details the definition of answer relevancy and faithfulness so that they can accurately compute the metrics by hand. Annotators were asked to rate the proportion of relevant facts as a proxy for answer relevancy, and the proportion of faithful facts as a proxy for faithfulness.

Human performance on GroUSE. Seven annotators were asked to assess the relevancy, completeness, usefulness, and faithfulness of answers. Each annotator was tasked with evaluating all 16 answers for one or more questions, depending on their availability. If they annotated several sets, the samples of the sets annotated subsequent to the first were shuffled. They were provided with definitions of each metric adapted from GPT-4 prompts, along with the question, the references, a reference answer, and the answer to be evaluated. Each sample was annotated two times by different annotators and the performance was computed on the average of the pass rate of the two annotators. Table 5 shows the inter-agreement rate for each metric between the two annotators of each sample. The agreement rate is over 90%, which augments confidence about the human evaluation. All annotators are fluent English speakers, one of them is familiar with the evaluated task, three have general knowledge of RAG, while the remaining three had no prior knowledge of the subject.

All labelers consented to share their annotations.

D Prompt templates used for evaluating a grounded answer

Prompt engineering. We measured the performances of eleven models on GroUSE, half of them closed-source, the other half open-source. For GPT-4 (Achiam et al., 2023), Gemini 1.0 Pro (Team et al., 2023), Mixtral 8x7b (Jiang et al., 2024), Prometheus 2 7b and Prometheus 2 8x7b (Kim et al., 2024), we iterated on the prompts, making our best effort to achieve the best possible results on the training set of GroUSE. These engineered prompts are then used to test the other models: the GPT-4 prompt is used for the whole GPT family and the Mixtral 8x7b prompt is used for Mixtral 8x22b (Team, 2024) and the Llama 3 models (MetaAI, 2024). To engineer a prompt, we begin with a basic instruction and evaluate how many tests in the training set it passes. We then qualitatively analyze the errors and craft a new prompt aimed at eliminating those errors. This iterative process continues until all tests pass or further progress becomes challenging. The amount of prompt tested for each model is visible Figure 7.

Prompt template. Each metric is evaluated with a separate prompt specifying its definition, however all prompts share the same template. We always ask the model to rate two answers, the first one being the ground truth and the second one being the answer we truly seek to evaluate: even without specifying that the first answer is the ground truth, this gives the evaluator model a point of comparison. The prompt format is as follow:

- Task introduction: Brief explanation of the grounded question answering task, and the expected citation format
- Evaluation instructions:
  - Context explanation: The model is required to assign a score to two answers.
  - Description of the metric: criteria to take into account to evaluate it, a rating scale detailing what each note entails, and a step by step explanation of the reasoning to follow.
  - Presentation of the architecture of the JSON expected as an answer: The JSON keys include chain-of-thought keys specific to the metric being evaluated (to compel the model to adhere to the reasoning steps), a free-form justification field, and the assignment of the score. The chain-of-thought keys include a boolean indicating whether the situation is adversarial or not in the **Answer relevancy** and **Usefulness** prompts, while the **Faithfulness** prompts asks for a sentence by sentence analysis, building on Chern et al. (2023); Min et al. (2023); Es et al. (2023). All these fields are repeated twice, once for each answer to evaluate. This step is absent in the PROMETHEUS 2 prompts as the output of the model is imposed.
- Sample: The query and references.
- The two answers to evaluate: The first one is always the ground truth, even though we never specify it in the prompt. The second one is the real answer we want to evaluate, and in practice we only look at the evaluation score of the second answer.

The prompts used for GPT-4 are available on Figures 8 to 11.

Ablation. We conduct an ablation experiment by measuring GPT-4's performance on GroUSE with different prompts: removing the ground truth and having the model rate only one answer, removing the justification field, and removing the chain-of-thought field. The results are shown in Table 6: the best agreement rates are obtained for the prompt without the justification, nonetheless removing the ground truth or the chain-of-thought lowers the performances.

Inter-Agreement Rate per Metric						Total
Answer relevancy	*Completeness*	*Usefulness*	*Faithfulness*	*Positive Acceptance*	*Negative Rejection*	Inter-Agreement rate
91.67	88.19	96.52	90.97	93.06	93.75	92.36

Table 5: Percentage of agreements between annotators when evaluating human performance.

	Agreement rate of metrics						Total
	Answer relevancy	*Completeness*	*Usefulness*	*Faithfulness*	*Positive acceptance*	*Negative rejection*	test pass rate
GPT-4	92.36	84.72	**100.0**	**93.75**	92.36	92.36	92.59
w/o ground truth	**93.75**	84.72	98.61	90.97	31.25	31.25	71.76
w/o justification	91.67	**88.89**	**100.0**	92.36	**98.61**	**98.61**	**95.02**
w/o chain of thought	90.28	84.72*	98.61	91.67	92.36	92.36	91.67

Table 6: Percentage of tests passed for different prompts. The highest score in each column is highlighted in bold. The completeness base prompt did not involve any chain of thought, so the reported result is the same as with the base prompt for this ablation, as marked by an asterisk.

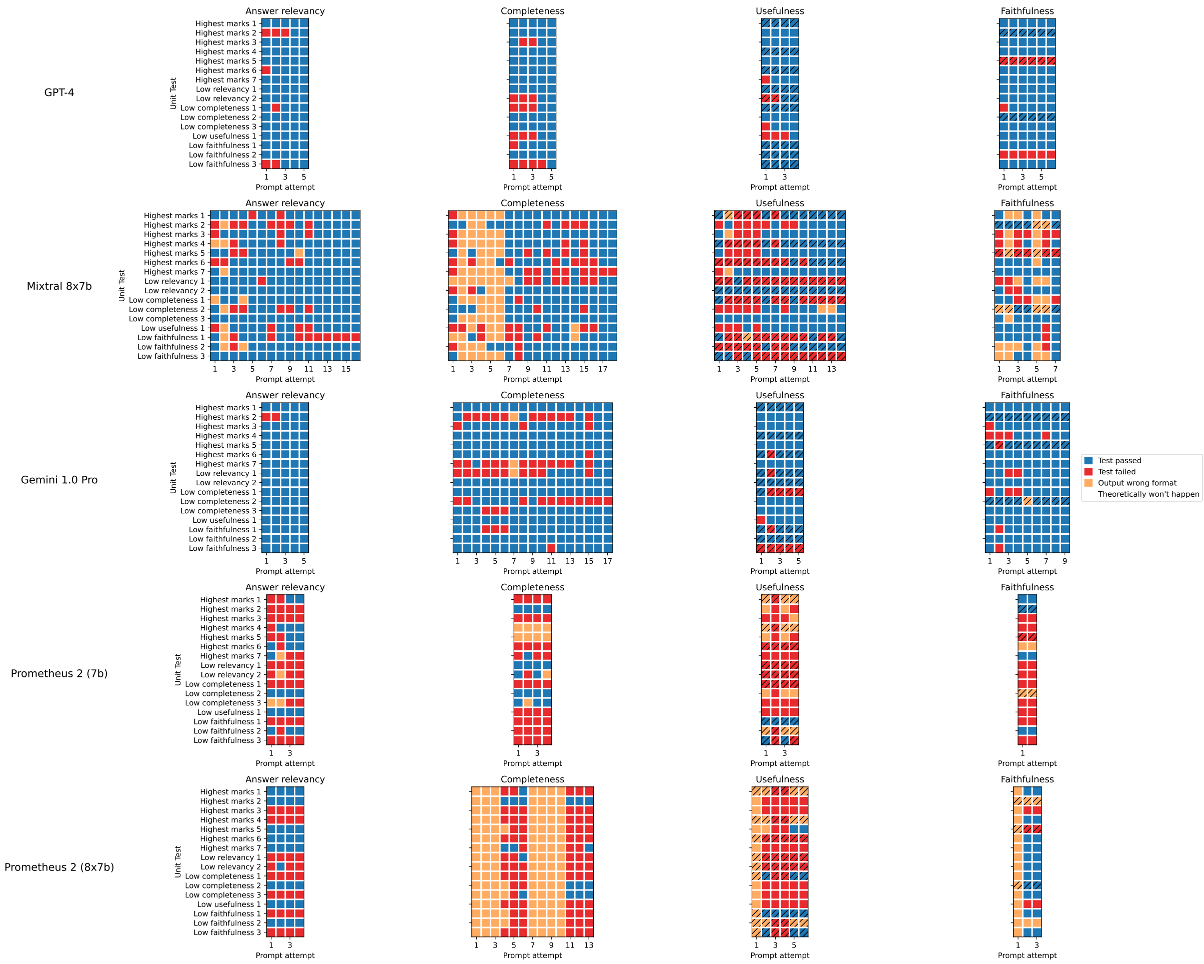

Figure 7: Matrixes representing the amount of prompt engineering, for each metric and each model. Each column in a matrix represents the results of one prompt on the training set of GroUSE, the number of column thus represents the number of prompts tested for a given model and metric.

```
[TASK]
Task: Grounded Question Answering
Based solely on the content of the references, the objective is to generate a response to the user's query. Each statement must be followed by the reference of the source passage, in the format [i] where i is the number of the reference. If no passage seems relevant, the answer should begin with "No document seems to precisely answer your question" and may be supplemented with related sourced information.
[/TASK]
[EVALUATION INSTRUCTIONS]
I will provide you with two answers, numbered 1 and 2, each containing a response to the user request.
I want you to assign to each answer a relevancy grade between 1 and 5:
- Answer relevancy evaluates if the content of the answer accurately responds to the user's question.
- The truthfulness of the information in the answer does not impact relevancy: even if information that appears false is contained in the answer, as long as this information is related to the request, then relevancy should not decrease. Remember that this information could come from references mentioning imaginary content that you are unaware of: the only thing to evaluate to assign the relevancy grade is therefore the adequacy between the information in the answer and the request, NOT their truthfulness.
- The absence of information in the answer does not impact relevancy, only the information contained in the answer is evaluated.
- Answer relevancy cannot be evaluated if the answer mentions that no document responds to the user request, it is then `null`, regardless of whether it contains other information or not.

Rating scale:
null - The answer asserts that no document precisely responds to the user request. Even if it provides additional \
information, whether appropriate or not, the relevancy remains `null`.
5 - The answer has excellent relevancy. All information provided in the answer is in line with the question \
and precisely answers the user request.
4 - The answer achieves good relevancy by providing relevant information to answer the user \
question. Some information indicated does not exactly answer the question, but remains in line with the request.
3 - The answer has average relevancy, it contains information that allows responding to the user request, \
but it also contains superfluous information, which was not necessary to answer the request.
2 - The answer shows low relevancy, with some elements related to the request, but the majority of \
the content is not in line with the question asked.
1 - The answer has very low relevancy, not answering the user's question at all. The \
content is largely inappropriate or off-topic, delivering no useful information for the request.

Before assigning each grade, you will check that the answer does not contain "No document responds...", if this is the case you must put a grade of `null`. If this is not the case, you will then analyze the adequacy between the request and the information contained in the answer.
Your response should be in JSON format, respecting the following format:
{
    "answer_1": {
        "answer_affirms_no_document_answers": X,
        "answer_relevancy_justification": "...",
        "answer_relevancy": Y
    },
    "answer_2": {
        "answer_affirms_no_document_answers": X,
        "answer_relevancy_justification": "...",
        "answer_relevancy": Y
    }
}
Where "..." is a string, X is a boolean, and Y is an integer between 1 and 5 or `null`.
[/EVALUATION INSTRUCTIONS]
[SAMPLE]
User request: [question]
[/SAMPLE]
[TO EVALUATE]
Answer 1: [ground_truth]
Answer 2: [prediction]
[/TO EVALUATE]
```

Figure 8: Prompt used for **Answer relevancy** metric with GPT models.

[TASK]
Task: Grounded Question Answering
Based solely on the content of the references, the objective is to generate a response to the user's query. Each statement must be followed by the reference of the source passage, in the format [i] where i is the number of the reference. If no passage seems relevant, the answer should begin with "No document seems to precisely answer your question" and may be supplemented with related sourced information.
[/TASK]
[EVALUATION INSTRUCTIONS]
I will provide you with two answers, numbered 1 and 2, each containing a response to the user request.
I want you to assign to each answer a completeness grade between 1 and 5:
- The only condition for an answer to be complete is the presence in it of at least all the information from the references that are relevant to the question asked.
- The presence of unrelated information in the answer does not impact completeness.
- The presence of information in the answer not from the references does not impact completeness.
- Possible errors in the sources citing the references do not impact completeness.
- Completeness cannot be evaluated if the references contain no information that can precisely answer the user request, in which case the grade takes the value `null`.

Rating scale:
null - The references contained no relevant information to precisely answer the user's question. In this case, there is no need to read the content of the answer to know that the grade is `null`.
5 - The answer is very complete, it contains all the relevant information from the references. No essential information is omitted, ensuring complete coverage of the question asked.
4 - The answer covers most of the relevant information in depth. It integrates the references satisfactorily, covering the majority of key points. Some details may be missing, but overall, the answer is substantial.
3 - The answer reasonably addresses a number of relevant aspects. It integrates part of the necessary information from the references. However, gaps remain, impacting the overall completeness.
2 - The answer only covers a minimal part of the relevant information. It misses several important information from the references.
1 - The answer covers none of the relevant information, all relevant information from the references has been omitted in the answer.

Before assigning each grade, you will always start by analyzing the information found in the references that are relevant to the user request. If there is no relevant information in the references, completeness must be `null`. If there are relevant information in the references, you will analyze which portion of this information is present or absent in the answers to evaluate the completeness grade. Your response should be in JSON format, respecting the following format:

```
{
    "answer_1": {
        "completeness_justification": "...",
        "completeness": X
    },
    "answer_2": {
        "completeness_justification": "...",
        "completeness": X
    }
}
```

Where "..." is a string, and X is an integer between 1 and 5 or `null`.
[/EVALUATION INSTRUCTIONS]
[SAMPLE]
List of references :
Reference 1: [reference 1]
Reference 2: [reference 2]
Reference 3: [reference 3]
User request: [question]
[/SAMPLE]
[TO EVALUATE]
Answer 1: [ground_truth]
Answer 2: [prediction]
[/TO EVALUATE]

Figure 9: Prompt used for **Completeness** metric with GPT models.

```
[TASK]
Task: Grounded Question Answering
Based solely on the content of the references, the objective is to generate a response to the user's query. Each statement must be followed by the reference of the source passage, in the format [i] where i is the number of the reference. If no passage seems relevant, the answer should begin with "No document seems to precisely answer your question" and may be supplemented with related sourced information.
[/TASK]
[EVALUATION INSTRUCTIONS]
I will provide you with two answers, numbered 1 and 2, each containing a response to the user request.
I want you to assign to each answer a usefulness grade of 0 or 1:
- Usefulness is only evaluated when the answer says that no document precisely answers the user's question, but it still provides information related to the question.
- Usefulness measures how interesting the related information is to know for the user, given that there is no answer in the references.
- If the answer responds to the user request, usefulness must be `null`.
- If the answer indicates that no document responds to the user request, without adding other information, usefulness must be `null`.

Rating scale:
null - (The answer responds to the user request) OR (the answer does not answer the user's question AND does not provide any related information).
1 - The related information is generally related to the question and adds value to the general understanding of the topic.
0 - The related information is completely off-topic with respect to the question asked.

Before assigning each grade, you will start by verifying that the answer indeed asserts "No document responds...", then you will check that the answer contains related information in addition to this assertion. If one of these two conditions is `false` then usefulness must be `null`.
If both conditions are indeed true, then you will analyze the usefulness of having added this related information to evaluate the usefulness grade. Your response should be in JSON format, respecting the following format:
{
  "answer_1": {
    "answer_affirms_no_document_answers": X,
    "answer_contains_related_information": X,
    "usefulness_justification": "...",
    "usefulness": Y
  },
  "answer_2": {
    "answer_affirms_no_document_answers": X,
    "answer_contains_related_information": X,
    "usefulness_justification": "...",
    "usefulness": Y
  }
}
Where "..." is a string, X is a boolean, and Y is an integer that is 0 or 1 or `null`.
[/EVALUATION INSTRUCTIONS]
[SAMPLE]
User request: [question]
[/SAMPLE]
[TO EVALUATE]
Answer 1: [ground_truth]
Answer 2: [prediction]
[/TO EVALUATE]
```

Figure 10: Prompt used for **Usefulness** metric with GPT models.

[TASK]
Task: Grounded Question Answering
Based solely on the content of the references, the objective is to generate a response to the user's query. Each statement must be followed by the reference of the source passage, in the format [i] where i is the number of the reference. If no passage seems relevant, the answer should begin with "No document seems to precisely answer your question" and may be supplemented with related sourced information.
[/TASK]
[EVALUATION INSTRUCTIONS]
I will provide you with two answers, numbered 1 and 2, each containing a response to the user request.
I want you to assign to each answer a boolean faithfulness grade. An answer is faithful if:
- Each statement made by the answer is followed by a source indicating the reference from which it is drawn.
- The information preceding the source is indeed from the corresponding reference.
- The information preceding the source is in agreement with the corresponding reference, and does not assert facts different from those indicated in the reference.
In all other cases, the response is considered non-faithful.
Faithfulness is also considered non-measurable if the answer asserts that no document responds to the question, and it does not provide any related information, it is then `null`.

Rating scale:
null - The answer asserts that no document responds to the question, and does not provide any related information.
1 - All sentences in the answer cite their sources, and are in agreement with the cited sources.
0 - At least one sentence in the response does not cite its sources, or cites a wrong source, or modifies the content from the references, or asserts something that is not supported by the cited references.

Before assigning each grade, you will start by verifying that the answer does not only assert "No document responds...", without any other information. If this is the case, then faithfulness must be `null`. Otherwise, I want you to analyze by explaining for each sentence, one after the other, if 1) a reference follows the sentence, 2) the reference following the sentence is correct, and 3) if the sentence does not distort or modify the content of the references. Your response should be in JSON format, respecting the following format:

```
{
    "answer_1": {
        "answer_only_asserts_no_document_answers": X,
        "content_analysis_sentence_by_sentence": [
            {
                "sentence": "...",
                "criterion_1": "...",
                "criterion_2": "...",
                "criterion_3": "..."
            },
            ...
        ],
        "faithfulness_justification": "...",
        "faithfulness": Y
    },
    "answer_2": {
        "answer_only_asserts_no_document_answers": X,
        "content_analysis_sentence_by_sentence": [
            {
                "sentence": "...",
                "criterion_1": "...",
                "criterion_2": "...",
                "criterion_3": "..."
            },
            ...
        ],
        "faithfulness_justification": "...",
        "faithfulness": Y
    }
}
```

Where "..." is a string, X is a boolean, and Y is either a boolean or `null`.
[/EVALUATION INSTRUCTIONS]
[SAMPLE]
List of references :
Reference 1: [reference 1]
Reference 2: [reference 2]
Reference 3: [reference 3]
[/SAMPLE]
[TO EVALUATE]
Answer 1: [ground_truth]
Answer 2: [prediction]
[/TO EVALUATE]

Figure 11: Prompt used for **Faithfulness** metric with GPT models.

E Detailed unit test results

Detailed performances of models on GroUSE are available on Figure 14 for closed models, and Figure 15 for open-source models. In these Figures, each square represents the result of one test.

These results were obtained through the OpenAI[5] and VertexAI[6] API for the GPT and Gemini models respectively. For Mixtral and Llama-3 models, the Fireworks AI[7] API was used. Prometheus 2 7b was deployed using TGI[8], and the inferences for Prometheus 2 8x7b were made using a model quantized with llama.cpp[9] (Q4_K_M quantization), and deployed with the same library. For all models, greedy decoding was used when available, else the smallest temperature allowed.

F French GroUSE Evaluation

Following a reviewer's advice, we translated GroUSE to French to study how the performance of Judge LLMs varies when the language of the question, contexts and answers is changed, while maintaining the evaluation instructions in English. Some prompts were slightly changed to adapt to the new dataset and achieve a satisfactory score on the training set of French unit tests. As shown in Table 7, the results are generally slightly lower compared to those presented in Table 3. This performance degradation may be attributed to the predominant English data seen during the training of these models and/or the linguistic mismatch between the instruction language and the context language. Despite this variation, OpenAI models continue to demonstrate superior performance, with GPT-4-turbo emerging as the top performer in terms of total pass rate. Among open-source alternatives, Mixtral 8x22b Instruct shows promising capabilities.

G Finetuning dataset constitution

Finetuning prompt format. Although Table 6 indicates that the best results are achieved without a justification, we opted to build the dataset of GPT-4 traces with one. This decision is supported by two main reasons: first, Mukherjee et al. (2023) show that a smaller model benefit more from GPT-4's traces if they include explanations of its reasoning. Second, the justification enhances the interpretability of the model's responses.

Models used for inference. Given the 1200 grounded QA statements, we used the following list of models to generate the predictions :

1. 412 answers were generated using a Llama 7b (Touvron et al., 2023) finetuned on a Grounded QA answering task.

2. 333 answers were generated using a Bloom 1b1 (Workshop et al., 2023) finetuned on a Grounded QA answering task.

3. 319 answers were generated using a Llama 13b (Touvron et al., 2023) finetuned on a Grounded QA answering task.

4. 136 answers were generated using a OpenHermes 2.5 Mistral-7B[10].

H Training and inference hyperparameters

The finetuning of the language model was conducted using the Meta-Llama-3-8B base model, employing an 8-bit quantization scheme to optimize memory efficiency. The model was trained to accommodate a sequence length of 7104 tokens, with sample packing enabled to maximize the utilization of input data. We utilized the LoRA (Low-Rank Adaptation) (Hu et al., 2022) technique using an adapter with parameters set to $r = 32$, $\alpha = 16$, and a dropout rate of 0.05.

Training was performed with a batch size of 64 over the course of three epochs, which took 2 hours on one A100 PCIe with 80GB of VRAM. The optimization process employed the AdamW (Loshchilov and Hutter, 2019) algorithm with an 8-bit implementation. A cosine learning rate scheduler was used, with a learning rate of 2.10^{-4} and 10 warmup steps. Two other trainings were conducted with learning rates 2.10^{-3} and 2.10^{-5}, but the results on GroUSE and alignment measures were less promising.

The inferences of the trained model were then conducted using greedy decoding.

[5] https://openai.com/index/openai-api/

[6] https://cloud.google.com/vertex-ai/docs/reference

[7] https://fireworks.ai/

[8] https://huggingface.co/docs/text-generation-inference/

[9] https://github.com/ggerganov/llama.cpp

[10] https://huggingface.co/teknium/OpenHermes-2.5-Mistral-7B

		Agreement rate of metrics						Total test pass rate
		Answer relevancy	*Completeness*	*Usefulness*	*Faithfulness*	*Positive acceptance*	*Negative rejection*	
Closed-source	**GPT-4**	**93.06**	83.33	**100**	91.67	92.36	92.36	92.12
	GPT-4o	89.58	85.42	99.31	**93.75**	90.97	90.97	91.67
	GPT-4-turbo	90.28	**88.19**	99.31	89.58	**95.14**	**95.14**	**92.94**
	GPT-3.5-turbo	85.42	50.69	80.56	66.67	70.83	64.58	69.79
	Gemini 1.0 Pro	72.22	55.55	84.73	47.22	79.17	80.56	69.91
Open-source	**Mixtral 8x7b Instruct**	77.08	52.78	83.33	68.75	78.47	74.31	72.45
	Mixtral 8x22b Instruct	88.89	73.61	99.31	86.11	86.81	83.33	86.34
	Llama-3.1 70b Instruct	88.19	61.11	97.92	75.00	84.03	81.94	81.37
	Llama-3.1 8b Instruct	62.50	29.17	86.11	68.06	63.89	65.28	62.62

Table 7: Percentage of tests passed for various models on French samples. They were evaluated with English instruction prompts following the Figure 5 pipeline. The highest score in each column is highlighted in bold.

I Finetuning results

Figure 13 shows a detailed comparison of the results between the Llama-3 8b before and after finetuning.

J Ablation: balancing the training dataset to reduce judgement biases

Dataset balance. To ensure the dataset encompassed a wide range of answer qualities, we utilized a diverse set of models to generate answers to the grounded QA statements. However, upon evaluating these answers, we observed certain GPT-4 biases in the distribution of marks: notably, a scarcity of score 2 for **Answer Relevancy**, and an overabundance of scores 1 and 5 for **Completeness**, as illustrated in the first row of Figure 12. To avoid propagating these biases in the finetuned model, we kept on predicting answers until the dataset seemed balanced enough, trying to select models with intermediate performances to produce answers of average quality and fill the gaps. The final balanced dataset was built choosing the 1400 answers which best harmonized the metrics among 4k evaluated grounded QA answers, resulting in the distribution shown in the second row of Figure 12.

Impact of training dataset imbalance. To assess the impact of dataset debiasing, we trained a model on the balanced dataset: this model will hereafter be referred to as the *balanced model*, as opposed to the model trained on the naive dataset, named the *unbalanced model*.

The evaluations of the balanced model closely mirror those of the unbalanced model. The grades of both models on the test set have an exact match of 58% for **Answer relevancy** and 63% for **Completeness**. Additionally, the Spearman correlation between the models for these metrics are 76% and 82%, respectively. The results on GroUSE and measured alignments are also close, with the unbalanced model showing a slightly higher correlation with GPT-4, while the balanced model performed marginally better on unit tests. Qualitative analysis of the balanced model's predicted marks reveals a persistent lack of intermediate scores. Overall, the debiasing process did not yield the anticipated improvements.

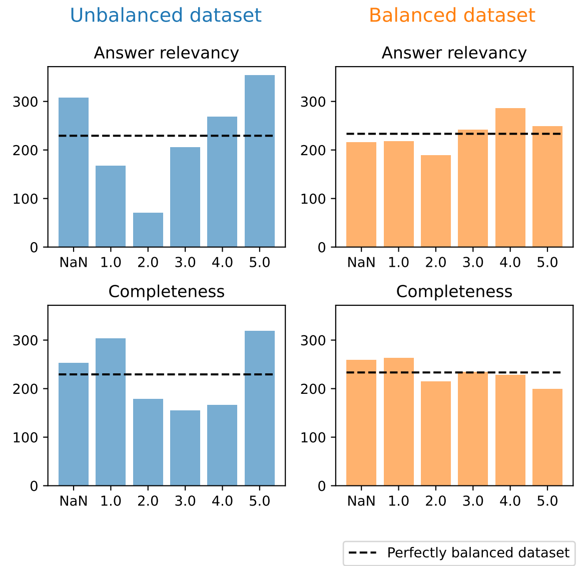

Figure 12: Comparison of the first and last dataset obtained during the iterative process of debiasing.

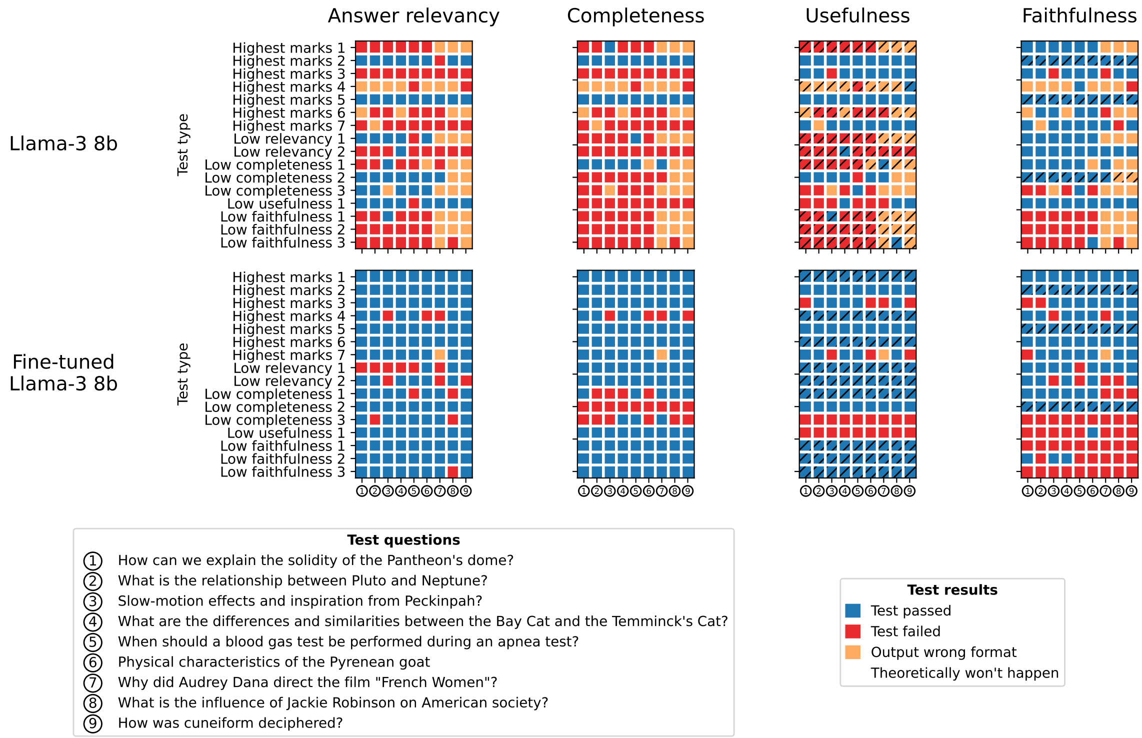

Figure 13: Comparison of unit tests results before and after finetuning of the Llama 3 8b model. Each matrix represents the performance of one model on a specific metric. Orange squares represent instances where the model's output did not adhere to the expected format, preventing score retrieval. Hatched squares denote LLM calls that the pipeline would skip if previous calls had returned the expected value (Figure 5).
Note that in this situation the four metrics were evaluated in a single prompt by the models, which explains the difference of results between the non finetuned Llama-3 8b depicted here and the Llama-3 8b results depicted in Figure 15.

	Agreement rate of metrics						Total
	Answer relevancy	*Completeness*	*Usefulness*	*Faithfulness*	*Positive acceptance*	*Negative rejection*	test pass rate
Finetuned Llama 3 8b *(Unbalanced dataset)*	**88.89**	81.94	81.25	52.78	91.67	**91.67**	81.37
Finetuned Llama 3 8b *(Balanced dataset)*	87.50	**83.33**	**81.94**	**61.81**	90.97	**92.36**	**82.99**

Table 8: Percentage of tests passed for balanced and unbalanced model. The highest score in each column is highlighted in bold.

	Spearman correlation		F1-score			
	Answer relevancy	*Completeness*	*Usefulness*	*Faithfulness*	*Positive acceptance*	*Negative rejection*
Finetuned Llama-3 8b *(Unbalanced dataset)*	0.62	0.52	0.32	0.57	0.65	0.73
Finetuned Llama-3 8b *(Balanced dataset)*	0.62	0.57	0.41	0.57	0.79	0.74

Table 9: Alignment with the ground truth (GPT-4) evaluations on the test set.

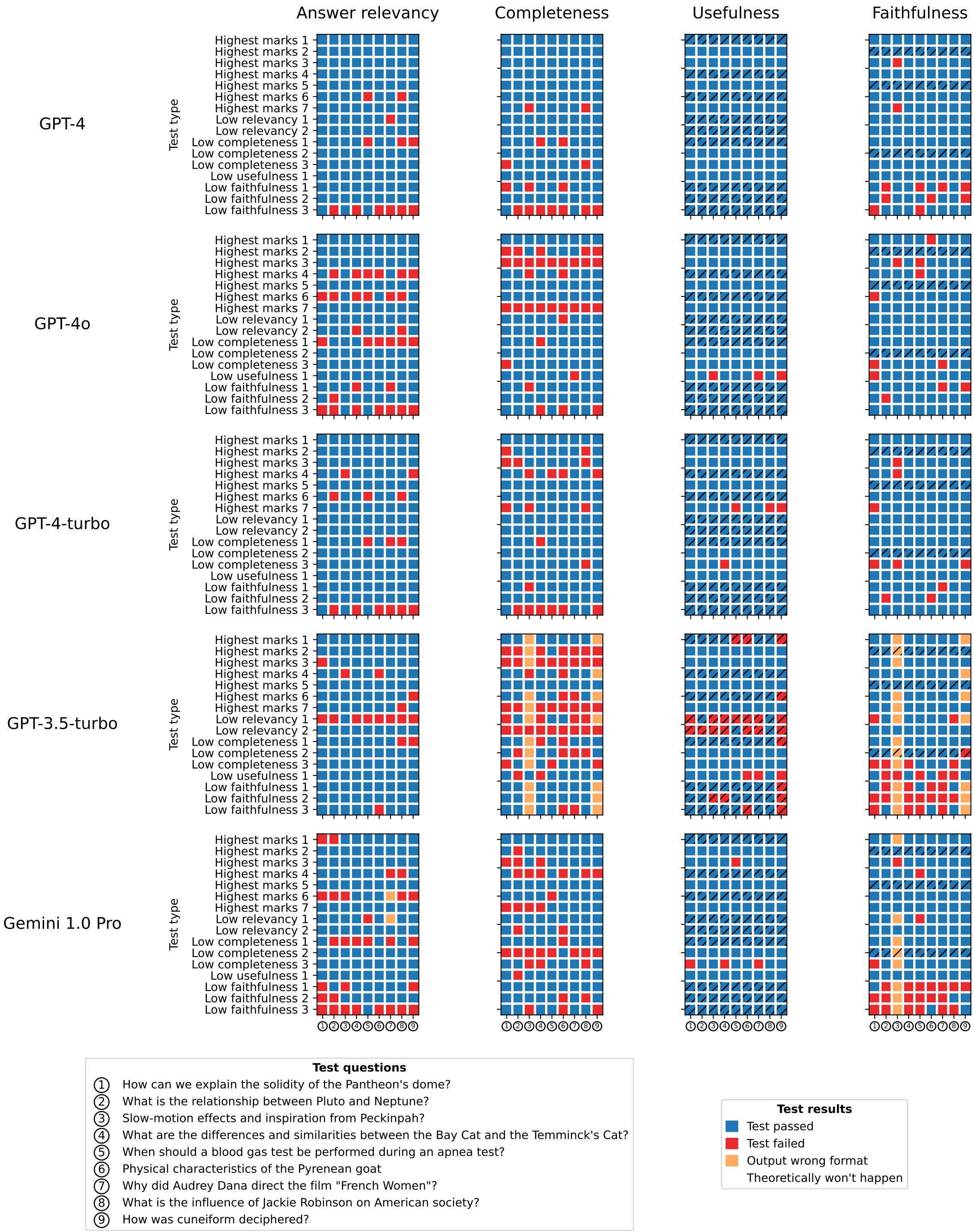

Figure 14: Detailed unit tests results for closed-source models. Each matrix represents the performance of one model on a specific metric. Orange squares represent instances where the model's output did not adhere to the expected format, preventing score retrieval. Hatched squares denote LLM calls that the pipeline would skip if previous calls had returned the expected value (Figure 5).

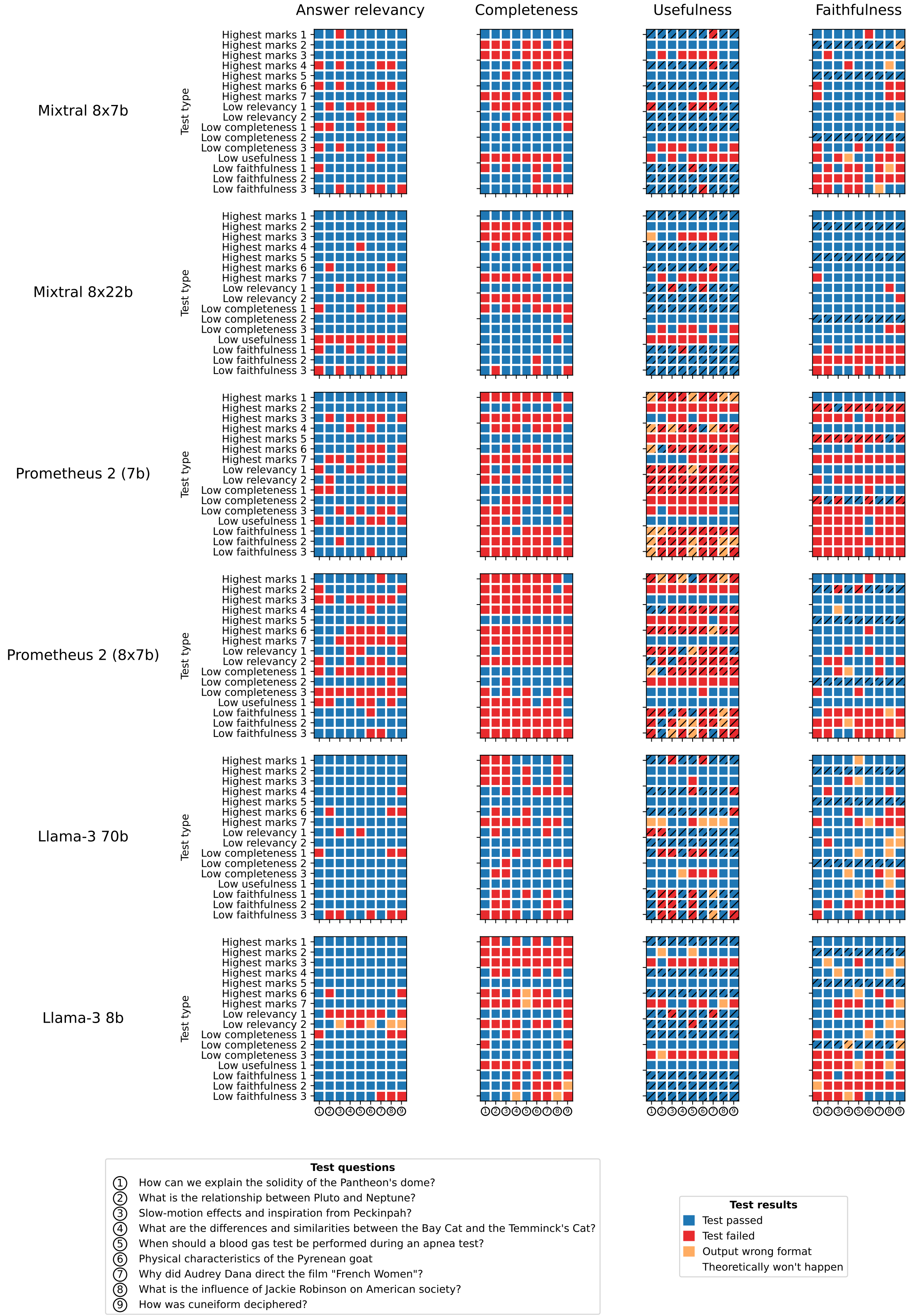

Figure 15: Detailed unit tests results for open-source models. Each matrix represents the performance of one model on a specific metric. Orange squares represent instances where the model's output did not adhere to the expected format, preventing score retrieval. Hatched squares denote LLM calls that the pipeline would skip if previous calls had returned the expected value (Figure 5).