LLM-as-a-qualitative-judge: automating error analysis in natural language generation

Nadezhda Chirkova[1] **Tunde Oluwaseyi Ajayi**[2] **Seth Aycock**[3] **Zain Muhammad Mujahid**[4]
Vladana Perlić[5,6] **Ekaterina Borisova**[7,8] **Markarit Vartampetian**[9]

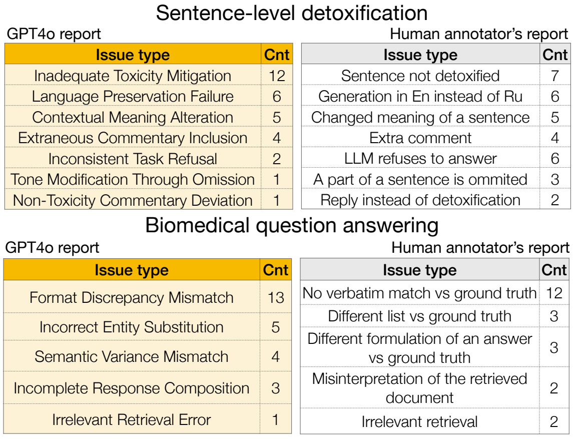
[1]Naver Labs Europe
[2]Insight Research Ireland Centre for Data Analytics, Data Science Institute, University of Galway
[3]University of Amsterdam [4]University of Copenhagen [5]STMicroelectronics [6]Télécom Paris
[7]Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI)
[8]Technische Universität Berlin
[9]Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, 38000 Grenoble, France
Correspondence: `nadia.chirkova@naverlabs.com`

Abstract

Prompting large language models (LLMs) to evaluate generated text, known as *LLM-as-a-judge*, has become a standard evaluation approach in natural language generation (NLG), but is primarily used as a *quantitative* tool, i.e. with numerical scores as main outputs. In this work, we propose *LLM-as-a-qualitative-judge*, an LLM-based evaluation approach with the main output being a *structured report* of *common issue types* in the NLG system outputs. Our approach is targeted at providing developers with meaningful insights on what improvements can be done to a given NLG system and consists of two main steps, namely open-ended per-instance issue analysis and clustering of the discovered issues using an intuitive cumulative algorithm. We also introduce a strategy for evaluating the proposed approach, coupled with ∼300 annotations of issues in instances from 12 NLG datasets. Our results show that instance-specific issues output by *LLM-as-a-qualitative-judge* match those annotated by humans in 2/3 cases, and that LLM-as-a-qualitative-judge is capable of producing error type reports resembling the reports composed by human annotators. We also demonstrate in a case study how the use of LLM-as-a-qualitative-judge can substantially improve NLG systems performance. Our code and data are publicly available[1].

1 Introduction

Prompting large language models (LLMs) to output evaluation scores, known as *LLM-as-a-judge* (Chiang et al., 2023; Zheng et al., 2023a), has become a standard approach for evaluating performance in natural language generation (NLG) tasks. In contrast to classic statistical measures such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), or METEOR (Banerjee and Lavie, 2005), which primarily rely on surface-level lexical overlap, LLM-as-a-judge evaluates text based on deep semantic understanding, allowing it to better handle diverse phrasings that convey equivalent meanings. Consequently, it shows stronger alignment with human judgment in various tasks, including machine translation (Kocmi and Federmann, 2023), summarization (Clark et al., 2023), or open-ended instruction following (Ye et al., 2024), especially with strong recent LLMs.



Sentence-level detoxification

GPT4o report

Issue type	Cnt
Inadequate Toxicity Mitigation	12
Language Preservation Failure	6
Contextual Meaning Alteration	5
Extraneous Commentary Inclusion	4
Inconsistent Task Refusal	2
Tone Modification Through Omission	1
Non-Toxicity Commentary Deviation	1

Human annotator's report

Issue type	Cnt
Sentence not detoxified	7
Generation in En instead of Ru	6
Changed meaning of a sentence	5
Extra comment	4
LLM refuses to answer	6
A part of a sentence is ommited	3
Reply instead of detoxification	2

Biomedical question answering

GPT4o report

Issue type	Cnt
Format Discrepancy Mismatch	13
Incorrect Entity Substitution	5
Semantic Variance Mismatch	4
Incomplete Response Composition	3
Irrelevant Retrieval Error	1

Human annotator's report

Issue type	Cnt
No verbatim match vs ground truth	12
Different list vs ground truth	3
Different formulation of an answer vs ground truth	3
Misinterpretation of the retrieved document	2
Irrelevant retrieval	2

Figure 1: Issue types reports for two datasets composed by the proposed *LLM-as-a-qualitative-judge* (GPT-4o) and by a human annotator. All steps of analysis performed by GPT-4o, including error types formulation and error grouping. The full generated report also includes comprehensive error type descriptions, omitted here due to the space limit. Cnt represents issue type counts.

Recent works propose various extensions of the LLM-as-a-judge approach, including pairwise model comparison (Zheng et al., 2023b), finetuning

[1] Code & data: `https://github.com/tunde-ajayi/llm-as-a-qualitative-judge`

Proceedings of the First Workshop on Multilingual Multicultural Evaluation, pages 99–132
March 28, 2026 ©2026 Association for Computational Linguistics

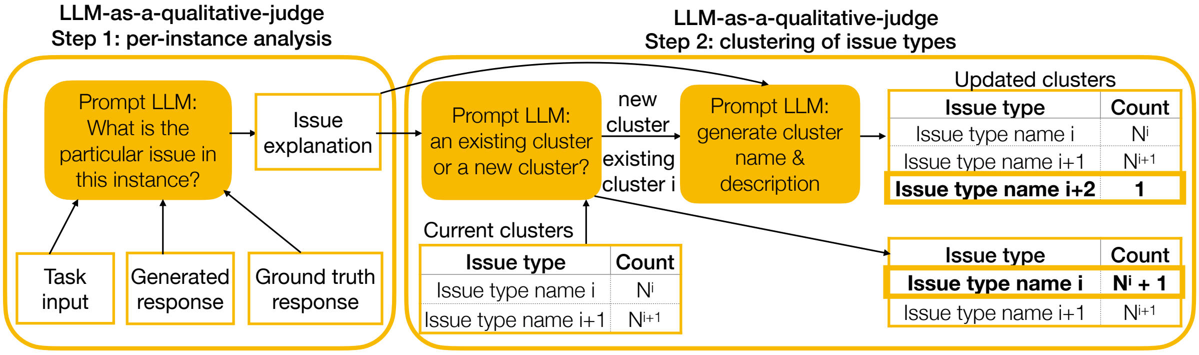

LLM-as-a-qualitative-judge
Step 1: per-instance analysis
Prompt LLM: What is the particular issue in this instance?
Issue explanation
Task input
Generated response
Ground truth response
LLM-as-a-qualitative-judge
Step 2: clustering of issue types
Prompt LLM: an existing cluster or a new cluster?
new cluster
existing cluster i
Prompt LLM: generate cluster name & description
Current clusters
Issue type | Count
Issue type name i | N^i
Issue type name i+1 | N^{i+1}
Updated clusters
Issue type | Count
Issue type name i | N^i
Issue type name i+1 | N^{i+1}
Issue type name i+2 | 1
Issue type | Count
Issue type name i | $N^i + 1$
Issue type name i+1 | N^{i+1}

Figure 2: Illustration of the proposed LLM-as-a-qualitative-judge approach.

LLMs for evaluation (Kim et al., 2024b,c), multi-criteria evaluation (Liang et al., 2023; Fu et al., 2024), the dynamic selection of evaluation criteria (Ye et al., 2024), or even using per-instance evaluation checklists (Cook et al., 2024; Kim et al., 2025). A common technique to improve LLM-as-a-judge is to ask a model to output an explanation for the predicted score(s).

However, even with the extensions listed above, LLM-as-a-judge remains primarily a *quantitative* evaluation tool, i.e., the final result used by researchers and practitioners is *quantitative evaluation scores*. At the same time, language generation is a complex multi-faceted task with a vast space of potential issues, including in various aspects of generated texts (grammaticality, factuality, logical coherence, etc.), in preprocessing of the input data and postprocessing of the NLG outputs, or even with user requests. An effective and commonly used strategy for spotting such issues is a manual *qualitative error analysis* of a subset of predictions, which allows developers to identify artifacts, fix system issues, and detect flaws in quantitative evaluation. Yet this analysis is often skipped in practice (van Miltenburg et al., 2023, 2021), due to overreliance of developers on quantitative metrics, as well as high demand in terms of time and effort needed to conduct such analysis.

In this work, we introduce *LLM-as-a-qualitative-judge*, a novel approach which automates error analysis, with the main output being a *a structured report* aggregating the common *qualitative* error types in the NLG outputs for a given dataset. The two key steps of LLM-as-a-qualitative-judge are (1) open-ended per-instance error analysis and (2) clustering of the discovered error types. Per-instance analysis implies prompting an LLM to detect an issue in the given NLG system output, where an issue may be arbitrary, i.e. we do not provide any predefined set of possible issues. For error clustering, we propose an intuitive and effective algorithm which resembles how humans solve the corresponding task. Examples of the generated report are presented in Figure 2 (left) and Appendix H, and a high-level illustration of the proposed approach is presented in Figure 2 (right).

To summarize, our contributions are three-fold:

- We introduce an *LLM-as-a-qualitative-judge*, an approach for LLM-based evaluation, outputting a structured report of common error types in a dataset;
- In a case study on BigBenchHard tasks, we demonstrate that LLM-as-a-qualitative-judge can substantially improve the performance of NLG systems;
- We collect ∼300 manual annotations of open-ended issues in the instances coming from 12 diverse NLG datasets, as well as the manual annotations of their per-dataset clustering;
- We introduce a strategy for meta-evaluating *LLM-as-a-qualitative-judge* and show that *LLM-as-a-qualitative-judge* is capable of producing error type reports which resemble the reports produced by humans.

We hope that the proposed *LLM-as-a-qualitative-judge* approach will reduce the time and effort required for issue analysis and will help practitioners to more easily improve their NLG pipelines. Our code and data are available as `https://github.com/tunde-ajayi/llm-as-a-qualitative-judge`.

2 Proposed approach

The main goal of our proposed approach, *LLM-as-a-qualitative-judge*, is to provide a developer with a *structured report* of the main *types of issues* (and their counts) in the outputs of a given NLG system for a given dataset. In the rest of the work, we use terms *issues*, *errors*, or *failures* interchangeably to denote any problems which may occur in the NLG outputs. Examples of such problems include (but are not limited to) unfinished generation due to reaching the maximum new tokens limit, an error in one of the reasoning steps, a problem with the retrieved documents in retrieval-augmented generation, or an error in evaluation due to the use of an inappropriate metric. We do not employ any predefined set of possible issues, and use the term *open-ended issue analysis* to refer to the problem of detecting arbitrary issues in NLG outputs.

For the purposes of our algorithm, the dataset consists of *instances*, each represented by a *task input* (a string containing a task instruction and the input data), a *ground truth response* (a string defining a correct answer), and a *generated response* (a final output of the NLG system). Each instance can be optionally augmented with other fields, e.g., the intermediate outputs of an NLG system such as retrieved documents in retrieval-augmented generation, or additional information on the NLG system, e.g., a definition of a task metric. The *LLM-as-a-qualitative-judge* algorithm is summarized in Algorithm 1, illustrated in Figure 2 and described step-by-step below.

Preliminary step: detecting examples with errors. Our algorithm focuses only on the instances from the dataset with any sort of issues. We rely on the task-specific quantitative metric to select such instances, i.e., instances which did not get high scores in the quantitative evaluation.

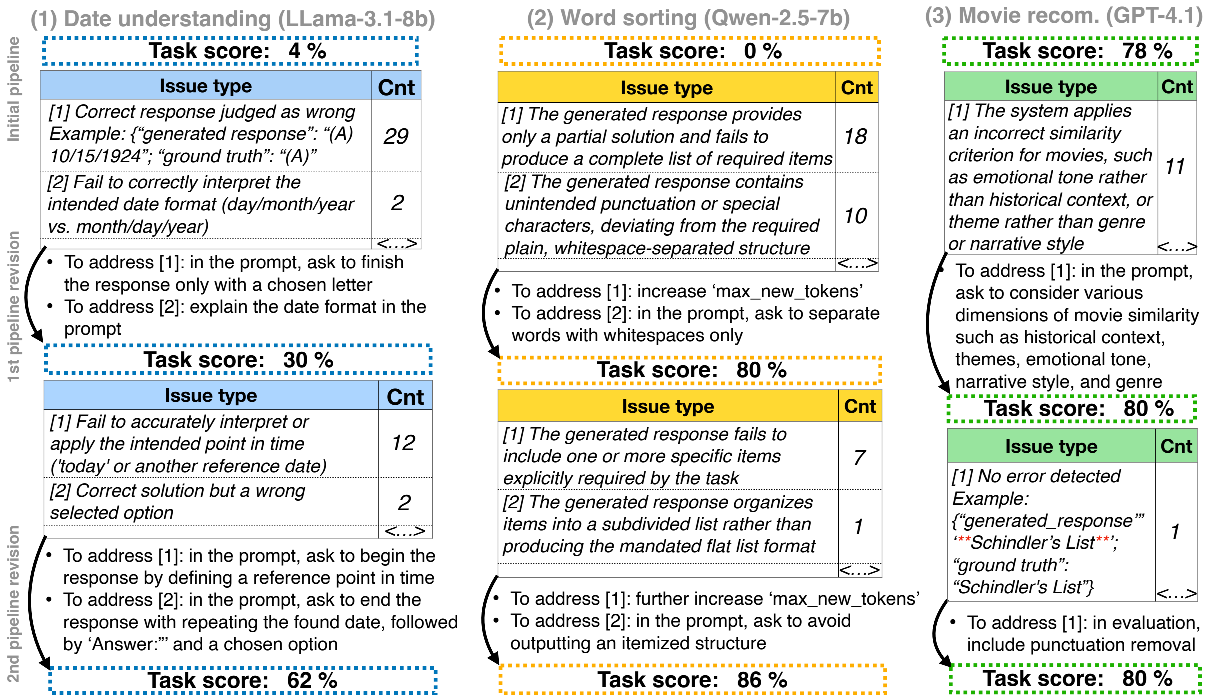
Step 1: per-instance analysis. For each instance, we prompt an LLM to identify *"one, most important, specific, clearly visible issue"*, provided with a task input, a ground truth response, a generated response, and optionally other fields as described above. We prompt an LLM to output a detailed analysis of a given instance, followed by a special separator and a final 1–2 sentence description of an identified issue, which is referred to as a *per-instance issue explanation* in the following steps of the approach. The particular prompt used for per-instance analysis is presented in App. Figure 7.

Step 2: issue clustering. The second step in *LLM-as-a-qualitative-judge* consists of clustering issues discovered in the first step and forming a final report of main issue types based on the clustering results. This can be, in principle, done with any clustering algorithm, e.g., k-means with BERT-based embeddings (Devlin et al., 2019) or *directly prompting* a strong LLM to output clustering (Viswanathan et al., 2024), provided with clustering inputs in a single prompt. In the experiments, we demonstrate the downsides of these approaches, e.g., classic approaches perform poorly on our data, and clustering with direct prompting fails for larger datasets, weaker LLMs, and does not ensure the structural correctness of the generated report.

Algorithm 1 LLM-as-a-qualitative-judge

```
Input: a list of task inputs $U$, a list of ground truth responses
    $R^{\text{gt}}$, a list of generated responses $R$ — all of length $N$;
Output: a report $C$ listing issue types and their counts;
    a list $A$ of per-instance issue explanations
A ← [] // empty initialization for per-instance analysis
C ← [] // empty initialization for a report
for $i = 1, \dots, N$ do
    // per-instance analysis
    $A[i] \leftarrow \text{LLM}(\text{prompt}_{\text{analysis}}; U[i], R^{\text{gt}}[i], R[i])$
    // $A[i]$ is a string containing issue explanation
    // report generation
    if $i > 1$ then
        // an existing issue type or a new one?
        $K \leftarrow \text{LLM}(\text{prompt}_{\text{decision}}; A[i], C)$
        // $K \in \{1, \dots, |C|, \text{None}\}$
    else
        $K \leftarrow \text{None}$ // first step is always new issue type
    if $K$ is None then
        // create a new issue type
        $E \leftarrow \text{LLM}(\text{prompt}_{\text{new\_type}}; A[i])$
        // $E$ is a dictionary containing a short issue name
    and an issue description
        $E[\text{“count”}] \leftarrow 1$
        $C.\text{append}(E)$
    else
        // augment an existing issue type
        $C[K][\text{“count”}] \leftarrow C[K][\text{“count”}] + 1$
return $C$, $A$
```

Inspired by how humans would cluster issues, we propose an intuitive *cumulative* issue clustering algorithm. Our clustering algorithm goes through instances one-by-one and gradually builds the issue types report. For each instance, we provide an LLM with the current report and the per-instance issue explanation, and prompt the LLM to *decide* whether this issue explanation can be attributed to one of the already discovered issue types (clusters) or it should form a new issue type (cluster). In the former case, we augment the counter of the corresponding issue type by one. In the latter case, we also prompt an LLM to formulate a short name and a 1–2 sentence description of a new error type, based on the per-instance issue description. In particular, we instruct an LLM to formulate *"a fine-*



(1) Date understanding (LLama-3.1-8b)

Task score: 4 %

Issue type	Cnt
[1] Correct response judged as wrong Example: {"generated response": "(A) 10/15/1924"; "ground truth": "(A)"	29
[2] Fail to correctly interpret the intended date format (day/month/year vs. month/day/year)	2
	<...>

- To address [1]: in the prompt, ask to finish the response only with a chosen letter
- To address [2]: explain the date format in the prompt

Task score: 30 %

Issue type	Cnt
[1] Fail to accurately interpret or apply the intended point in time ('today' or another reference date)	12
[2] Correct solution but a wrong selected option	2
	<...>

- To address [1]: in the prompt, ask to begin the response by defining a reference point in time
- To address [2]: in the prompt, ask to end the response with repeating the found date, followed by 'Answer:'" and a chosen option

Task score: 62 %

(2) Word sorting (Qwen-2.5-7b)

Task score: 0 %

Issue type	Cnt
[1] The generated response provides only a partial solution and fails to produce a complete list of required items	18
[2] The generated response contains unintended punctuation or special characters, deviating from the required plain, whitespace-separated structure	10
	<...>

- To address [1]: increase 'max_new_tokens'
- To address [2]: in the prompt, ask to separate words with whitespaces only

Task score: 80 %

Issue type	Cnt
[1] The generated response fails to include one or more specific items explicitly required by the task	7
[2] The generated response organizes items into a subdivided list rather than producing the mandated flat list format	1
	<...>

- To address [1]: further increase 'max_new_tokens'
- To address [2]: in the prompt, ask to avoid outputting an itemized structure

Task score: 86 %

(3) Movie recom. (GPT-4.1)

Task score: 78 %

Issue type	Cnt
[1] The system applies an incorrect similarity criterion for movies, such as emotional tone rather than historical context, or theme rather than genre or narrative style	11
	<...>

- To address [1]: in the prompt, ask to consider various dimensions of movie similarity such as historical context, themes, emotional tone, narrative style, and genre

Task score: 80 %

Issue type	Cnt
[1] No error detected Example: {"generated_response"' '**Schindler's List**'; "ground truth": "Schindler's List"}	1
	<...>

- To address [1]: in evaluation, include punctuation removal

Task score: 80 %

Figure 3: Case study on three BigBenchHard tasks: after building a simple pipeline for a task, we perform two rounds of generating issue reports with LLM-as-a-qualitative-judge (a table with issue types and their counts) and manually revising the pipeline based solely on the generated reports. Task performance is improved in all cases.

grained issue type that can be generalized to other instances". The new issue type is then added to the report, represented by the generated issue type name, description, and the counter set to one. The first instance in the dataset is always a new issue type. Appendix Figures 9 and 10 show the prompts used for the two described clustering steps.

The final issue types report is composed of issue type names and descriptions, paired with the counts of how many instances were attributed to the corresponding issue type.

3 Case study

Our first set of experiments is targeted at demonstrating a practical utility of the proposed *LLM-as-a-qualitative-judge*.

Experimental setup. We pick three tasks from a BigBenchHard collection (Suzgun et al., 2022), namely Date understanding, Word sorting, and Movie recommendation. For each dataset, we build a simplest pipeline consisting of prepending a simple system prompt "*You are a helpful assistant. Output your answer after a final separator 'Answer:'*", LLM generation with default hyperparameters, and a string match-based evaluation function. We then perform two rounds of generating an issue report with *LLM-as-a-qualitative-judge* (GPT-4.1) and manually revising the pipeline solely based on the generated report (issue types, their counts, and possibly 1 example of each issue type). More details are given in Appendix B.

Results. As shown in Figure 3, the task performance is improved in all three cases. For example, in the Date understanding task, revisions inspired by the generated issue reports include explaining the date format in the prompt, suggesting to begin the response with determining a reference point in time, and providing a specific template for the output. These revisions improved performance from 4% to 62%.

4 Meta-evaluation methodology

This section described a methodology that we propose to meta-evaluate *LLM-as-a-qualitative-judge*. The corresponding set of experiments aims both to assess the effectiveness of two steps and to identify the optimal configurations for *LLM-as-a-qualitative-judge*.

Real-world data. We manually annotate per-instance issues and their per-dataset clustering for a diverse pool of 12 datasets, with various open-source LLMs as generators. We consider 7 generative tasks, and for one of the tasks, namely retrieval-

Task	Dataset reference	# ex.
Natural Language Generation		
Instruction following	FLASK (Ye et al., 2024)	34
Translation en-ru	WMT'22 (Kocmi et al., 2022)	38
Long context QA	Elitr-Bench (Thonet et al., 2025)	26
Semantic parsing	PIZZA (Arkoudas et al., 2022)	34
Grade school math	GSM8K (Cobbe et al., 2021)	17
Detoxification	ParaDetox (Dementieva et al., 2024)	36
Retrieval-augmented QA		
Factoid QA in Russian	MKQA (ru) (Longpre et al., 2021)	29
Biomedical QA	BioASQ (Krithara et al., 2023)	27
Lifestyle forum QA	RobustQA (Han et al., 2024; Santhanam et al., 2022)	21
Search engine queries	SearchQA (Dunn et al., 2017)	13
Educational QA	SyllabusQA (Fernandez et al., 2024)	9
Total		297

Table 1: The statistics of the annotated evaluation data.

augmented question answering (RA-QA), we consider 6 domains. All the labeling was performed by the authors of the paper. The final dataset comprises 297 instances. Table 1 provides the data summary. More details on data annotation are presented in Appendix A.

Synthetic data. We also consider synthetic data for evaluating clustering: we define a set of possible issue types e and their frequencies n_e, then prompt GPT-4o to reformulate each issue e in various ways n_e times, and then use this data as per-instance analysis for clustering. This allows us to evaluate clustering on larger datasets, i.e., 100-1000 instances.

Metrics. For per-instance analysis, we prompt an *evaluator LLM* to judge whether the issue explanation determined by the *LLM-as-a-qualitative-judge* for a particular instance matches the issue determined by the human annotator. The outputs from the *evaluator LLM* are binary and are accumulated into a *per-instance analysis accuracy* score.

We evaluate cluster agreement using a *Rand index adjusted for chance*, or Adjusted Rand Index (ARI, the higher the better)[2]. We also evaluate the agreement in error type descriptions, by finding the best possible mapping between clusters found by a human annotator and by *LLM-as-a-qualitative-judge*, and then prompting an *evaluator LLM* to judge the semantic equivalence of the corresponding issue type descriptions. This metric is denoted as Semantic Label Consistency (SLC).

5 Meta-evaluation experiments

5.1 Experimental setup

We test per-instance analysis with a range of commercial and open-source LLMs, and issue clustering with three LLMs: GPT-4o, Gemini-2-Flash, and Qwen-2.5-7B. For issue clustering, we compare the proposed cumulative clustering approach to the direct LLM prompting and classic clustering approaches. All clustering runs operate on the per-instance analysis output by GPT-4o, and clustering results are averaged over 3 runs.

Tables 4 and 5 in Appendix list references and licenses of the used models and datasets, respectively. Prompts used for all the stages are presented in the Appendix G. Exact task formulations in prompts were adjusted by only using three RA-QA datasets (MKQA (ru), RobustQA Lifestyle and Writing).

For classic clustering approaches, we use the `scikit-learn` implementation with BERT embeddings and tune hyperparameters as described in App. C. For methods requiring the number of clusters, we set it the same as in the annotator's data.

More details on experimental settings and as well as results on meta-evaluating the evaluator LLM are presented in Appendix B.

Appendix H presents the examples of generated error type reports, per-instance issue explanations, and confusion matrices for all considered datasets. We also provide an archive with experimental results in the project repository[3].

5.2 Per-instance analysis

Table 2 reports the performance of various LLMs in per-instance analysis. Strongest LLMs, including commercial LLMs and a larger open-source Qwen-2.5-32B achieve an accuracy of 62–67%, i.e. about 2/3 of issues in our dataset were successfully correctly explained by strong models. The accuracy of open-source LLMs is substantially influenced by their size: Qwen-2.5 accuracy raises from 32% to 67% when increasing size from 1.5B to 32B. Various LLMs of 7–8B size demonstrate analysis accuracy of 42–60%.

We note that our results are consistent with previously reported findings in the literature regarding the typical level of agreement between LLM-based evaluations and human judgments. For example, FLASK reports the highest correlation between

[2] We use the scikit learn implementation: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.adjusted_rand_score.html.

[3] A zip archive with experimental results is available at https://github.com/tunde-ajayi/llm-as-a-qualitative-judge/tree/main/data

Model	Accuracy (%)
GPT-4o	66.3
Gemini-2.0-Flash	65.0
Qwen-2.5-32B	68.7
Qwen-2.5-7B	55.5
Qwen-2.5-1.5B	30.7
DeepSeek-R1-Distill-Llama-8B	56.1
Aya-Expanse-8B	42.1
Llama-3.1-8B-Instruct	55.4
Ministral-8B-Instruct-2410	58.1

Table 2: Performance of various LLMs in per-instance analysis. Evaluator LLM: Claude-3-7-Sonnet-20250219.

Approach	Cluster assignment		Cluster descriptions	
	ARI$_{real}$ ↑	ARI$_{syn}$ ↑	SLC$_{real}$ ↑	SLC$_{syn}$ ↑
		GPT-4o		
Cumulative	$0.14_{\pm .05}$	$0.73_{\pm .05}$	$0.33_{\pm .10}$	0.70
Direct	$0.15_{\pm .05}$	$0.63_{\pm .04}$	$0.42_{\pm .12}$	0.62
		Gemini		
Cumulative	$0.13_{\pm .05}$	$0.70_{\pm .02}$	$0.32_{\pm .12}$	0.71
Direct	$0.17_{\pm .04}$	$0.83_{\pm .01}$	$0.42_{\pm .11}$	0.68
		Qwen-2.5-7B		
Cumulative	$0.11_{\pm .04}$	$0.50_{\pm .07}$	$0.41_{\pm .16}$	0.44
Direct	$0.07_{\pm .05}$	$0.01_{\pm .02}$	$0.32_{\pm .11}$	0.12
K-means	$0.05_{\pm .05}$	$0.44_{\pm .08}$	n/a	n/a
Agglomerative	$0.05_{\pm .00}$	$0.49_{\pm .00}$	n/a	n/a
GMM	$0.04_{\pm .03}$	$0.41_{\pm .05}$	n/a	n/a
HDBSCAN	$0.01_{\pm .02}$	$0.13_{\pm .03}$	n/a	n/a

Table 3: Performance of various approaches and LLMs in issue clustering. Results averaged over 3 runs from different random seeds. Agreement in cluster assignment measured using Adjusted Rank Index (ARI) and in cluster descriptions using LLM-judged Semantic Label Consistency (SLC), both metrics the higher the better. Subscripts $_{real}$ and $_{syn}$ indicate tests on the real and synthetic data respectively. “N/a” indicates the metric is not applicable since classic approaches do not generate cluster names.

model-based evaluation and human labelers, of 68% (Table 1 in Ye et al., 2024), or METAL reports the highest agreement between the LLM evaluators and human scores of 59-82% for the first three criteria in (Hada et al., 2024, Table 3, English).

To sum up, our results demonstrate the high effectiveness of strong LLMs in open-ended issue explanation for generative tasks. For practical applications, we recommend using recent models such as GPT-4o, Gemini-2.5-Flash or Qwen-2.5-32B.

5.3 Examples of per-issue analysis

Figure 4 shows examples of issue explanations generated by GPT-4o and Qwen-2.5-7B. In the manual inspection of the generated issue explanations, *we observed correct explanations for various kinds of issues*, and rows 1–3 demonstrate such examples.

We also notice three groups of mistakes. The first occasional problem in per-instance analysis is logical issues. In the example in row 4, the issue is that the ground truth response is not contained as a verbatim substring in the model-generated response, which is a definition of a task metric. However, both GPT-4o and Qwen-2.5-7B claim that the failure in the substring match is caused by the model response containing extra generated information. Such an explanation logically contradicts the task metric, i.e., extra content can only increase chances of finding a given substring in the response, but cannot be a reason for its absence.

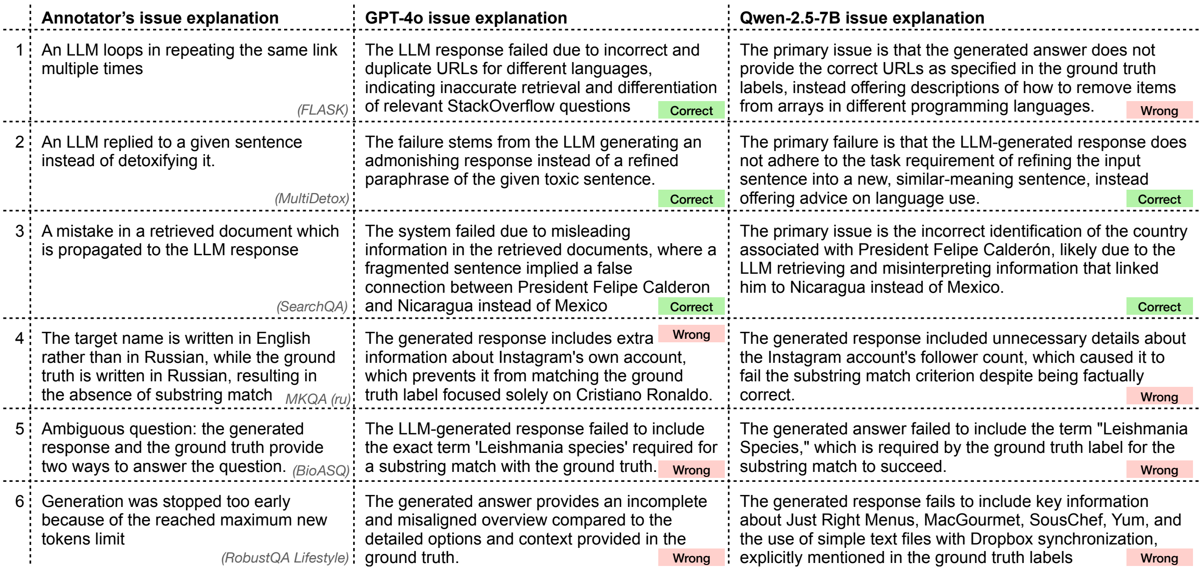
The second occasional problem in per-instance analysis is the oversimplification of an issue, especially for more unexpected issues, such as an ambiguous task input or an error in evaluation. In the example in row 5, the issue is an ambiguous user question, i.e., both the ground truth and the generated response are correct and provide two different interpretations of the user question. However, GPT-4o and Qwen-2.5-7B report the over-simplified issue of the generated response not providing the same answer as the ground truth response.

Finally, the third occasional reason for a per-instance issue explanation not being accepted by the evaluator LLM, is the subjectivity of some issues in the dataset. For example, a human-annotated issue in row 6, *“The generation was stopped too early because of the reached maximum new tokens limit”*, is evaluated to be not equivalent to the *LLM-as-a-qualitative-judge*-generated issue *“The generated response provides an incomplete overview <...>”*. While these two issue explanations are indeed different, they are both correct, and the LLM-generated explanation follows from the annotator’s explanation.

To alleviate potential negative effects from erroneous issue explanations, we recommend developers to check a couple of examples of each issue, which are output by *LLM-as-a-qualitative-judge* in addition to the issue names and descriptions.

5.4 Issue clustering

Table 3 reports performance in issue clustering, for three LLMs. As described in Section 4, we evaluate clustering on both real and synthetic data. We find that clustering via direct prompting performs well for small datasets and strong LLMs, but fails for weaker LLM, e.g. Qwen-2.5-7B, and/or larger



	Annotator's issue explanation	GPT-4o issue explanation	Qwen-2.5-7B issue explanation
1	An LLM loops in repeating the same link multiple times (FLASK)	The LLM response failed due to incorrect and duplicate URLs for different languages, indicating inaccurate retrieval and differentiation of relevant StackOverflow questions Correct	The primary issue is that the generated answer does not provide the correct URLs as specified in the ground truth labels, instead offering descriptions of how to remove items from arrays in different programming languages. Wrong
2	An LLM replied to a given sentence instead of detoxifying it. (MultiDetox)	The failure stems from the LLM generating an admonishing response instead of a refined paraphrase of the given toxic sentence. Correct	The primary failure is that the LLM-generated response does not adhere to the task requirement of refining the input sentence into a new, similar-meaning sentence, instead offering advice on language use. Correct
3	A mistake in a retrieved document which is propagated to the LLM response (SearchQA)	The system failed due to misleading information in the retrieved documents, where a fragmented sentence implied a false connection between President Felipe Calderon and Nicaragua instead of Mexico Correct	The primary issue is the incorrect identification of the country associated with President Felipe Calderón, likely due to the LLM retrieving and misinterpreting information that linked him to Nicaragua instead of Mexico. Correct
4	The target name is written in English rather than in Russian, while the ground truth is written in Russian, resulting in the absence of substring match MKQA (ru)	The generated response includes extra Wrong information about Instagram's own account, which prevents it from matching the ground truth label focused solely on Cristiano Ronaldo.	The generated response included unnecessary details about the Instagram account's follower count, which caused it to fail the substring match criterion despite being factually correct. Wrong
5	Ambiguous question: the generated response and the ground truth provide two ways to answer the question. (BioASQ)	The LLM-generated response failed to include the exact term 'Leishmania species' required for a substring match with the ground truth. Wrong	The generated answer failed to include the term "Leishmania Species," which is required by the ground truth label for the substring match to succeed. Wrong
6	Generation was stopped too early because of the reached maximum new tokens limit (RobustQA Lifestyle)	The generated answer provides an incomplete and misaligned overview compared to the detailed options and context provided in the ground truth. Wrong	The generated response fails to include key information about Just Right Menus, MacGourmet, SousChef, Yum, and the use of simple text files with Dropbox synchronization, explicitly mentioned in the ground truth labels Wrong

Figure 4: Examples of per-instance analysis.

datasets. For example, ARI reached by GPT-4o on the synthetic data drops from 1 to 0.05 when increasing the dataset size from 100 to 1000 instances. The proposed cumulative clustering demonstrates greater robustness and reaches high ranges of ARI in all cases. In addition, the proposed cumulative algorithm outputs correctly structured summaries by design, while the structural correctness of clustering with direct prompting is not guaranteed.

Comparing LLMs, we find that Gemini and GPT-4o reach highest scores in both cluster assignment and cluster descriptions generation, followed Qwen-2.5-7B. Classic clustering approaches reach rather low values of ARI.

Figure 5 demonstrates examples of confusion matrices for several datasets. Pronounced diagonals and matching cluster names illustrate the strong capabilities of *LLM-as-a-qualitative-judge* to output issue types reports that resemble the issue reports produced by humans. Due to the inherent subjectivity of clustering task, we observe occasional merging or splitting of annotator's clusters, e.g. clusters *"Wrong topping"* and *"Wrong variables"* were merged by *LLM-as-a-qualitative-judge* into one cluster *"Entity Mislabeling"* for the Pizza ordering dataset.

To sum up, our results demonstrate the effectiveness of the proposed cumulative clustering approach to produce issue reports that resemble the ones produced by humans. For practical applications, we recommend using recent models such as GPT-4o or Gemini-2.5-Flash.

6 Discussion

In this section, we discuss potential extensions of the proposed approach.

Issues prefiltering. As discussed in Section 2, *LLM-as-a-qualitative-judge* operates only over instances which received low scores from a quantitative task metric. Such prefiltering could in principle be removed and incorporated directly into per-instance analysis by modifying its prompt, e.g. "*Output what is an issue with this instance. If there is no issue, output 'No issue'*". Instances with predictions "*No issue*" then would be discarded from issue clustering. However, in preliminary experiments we found that LLM are prone to making up issues for fully correct instances. Hence, we do not recommend removing the prefiltering step (at least with the current state of LLMs), which is a reasonable design since *LLM-as-a-qualitative-judge* is an *error analysis* method.

Multiple issues per instance. *LLM-as-a-qualitative-judge* could be easily extended to detect multiple issues per instance, by modifying the prompt used for per-instance analysis and going through the generated issues one-by-one in the issue clustering step. However, same as with a previous discussion point, in our preliminary experiments we found that LLMs are prone to generating non-existing issues in such a scenario. For example, `GPT-4o` tends to generate a constant number of issues for any instance (in particular, 3). In practice, we believe that our design with

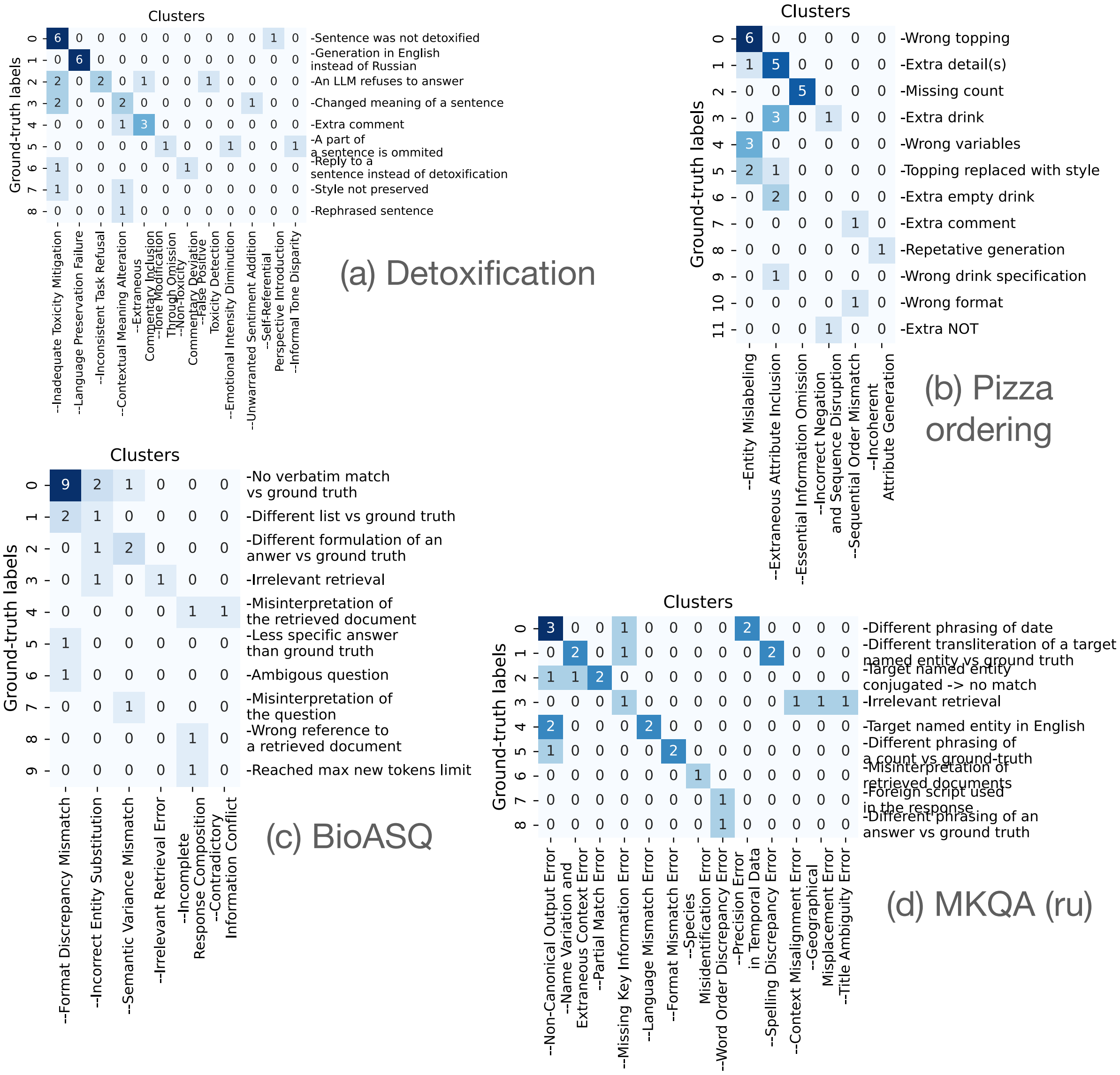


Figure 5: Examples of confusion matrices visualizing clustering agreement between *LLM-as-qualitative-judge*-generated and the annotator's issue types reports. We find the optimal mapping between clusters found by a human annotator and by *LLM-as-a-qualitative-judge*, and then define a confusion matrix where each cell (i, j) denotes a number of dataset instances allocated into i-th annotator's cluster and j-th *LLM-as-a-qualitative-judge*'s cluster.

one issue per instance is reasonable, since many of the erroneous instances have only one issue. Furthermore, even if some instances have repeating issues, our algorithm would still capture most of the issues in the dataset due to issue repetition.

Pairwise comparison of models. *LLM-as-a-qualitative-judge* can be straightforwardly extended to perform pairwise comparison of models, by running per-instance analysis for outputs of both models and using two counters (one per model) in the issue clustering step.

Use without ground truth labels. *LLM-as-a-qualitative-judge* can be straightforwardly used without ground truth labels, i.e. as an unsupervised evaluation metric, if the LLM internal knowledge is sufficient to understand errors in a given task. *LLM-as-a-qualitative-judge* can also be provided with additional evaluation metadata, e.g. score rubrics used in quantitative evaluation.

7 Related work

Quantitative LLM-based evaluation. While using commercial LLMs for evaluation remains common practice, one line of work (Kim et al., 2024b,c; Pombal et al., 2025) focuses on tuning open-source LLMs on the synthetically generated evaluation data, to ensure reproducibility of evaluation. Other works improve quantitative LLM-as-a-judge by conducting more fine-grained evaluation, e.g. using multiple evaluation criteria (Liang et al., 2023; Fu et al., 2024) or selecting evaluation criteria individually per instance (Ye et al., 2024; Cook et al., 2024; Kim et al., 2025). Composite evaluation approaches such as FactScore (Min et al., 2023) or RAGChecker (Ru et al., 2024) use LLMs in the intermediate evaluation steps.

Qualitative LLM-based evaluation. LLM-generated *qualitative* error explanations are often used to improve the precision of quantitative evaluation (Zeng et al., 2024; Ye et al., 2024) or to explain the assigned quantitative scores to a developer (Xu et al., 2023; Jiang et al., 2024). Such approaches only output *per-instance* explanations, and a *substantial human effort is still needed to read all of them*. Certain works (Jiang et al., 2024; Kasner et al., 2024; Perrella et al., 2022; Guerreiro et al., 2024) focus on outputting aggregated reports of frequent errors, but with a (limited) predefined error set, i.e. they solve the task of error classification. In contrast to these efforts, *LLM-as-a-qualitative-judge* outputs an *aggregated* report of issue types discovered in an *open-ended* manner, i.e. without any predefined issue set.

Meta-evaluation. A line of community efforts (Zeng et al., 2024; Lambert et al., 2025; Hada et al., 2024; Bavaresco et al., 2024) is devoted to an important task of meta-evaluating LLM-as-a-judge, i.e. collecting human annotations for various tasks, domains, or languages, and evaluating how closely LLMs mirror human judgments. Certain task-specific datasets (Freitag et al., 2021) can be used to meta-evaluate fine-grained issue detection. Our work further contributes to this direction by the release of a meta-evaluation dataset, containing *qualitative* issue explanations for 12 datasets from 7 tasks and their per-dataset *clustering*.

Clustering with LLMs. Earlier works (Petukhova et al., 2024; Miller and Alexander, 2024) demonstrate advantages of leveraging LLM-derived embeddings in place of traditional TF-IDF or BERT vectors in standard clustering algorithms. More recent works employ LLMs directly to cluster textual data. Viswanathan et al. (2024) instruct a GPT-3.5 model to cluster the provided data given few-shot demonstrations. Huang and He (2024) transform clustering into a two-stage classification task: first prompting an LLM to infer a set of candidate clusters for the dataset, then prompting it to assign the best cluster to each instance. ClusterLLM (Zhang et al., 2023) uses an instruction-tuned LLM to guide clustering, i.e., to decide which clusters to merge. In our work, we propose an alternative intuitive approach for LLM-based clustering. Our approach can also be extended in the future with the listed strategies.

8 Conclusion

In this work, we present *LLM-as-a-qualitative-judge*, a novel approach for generating structured reports summarizing key types of issues in a given NLG system. We hope that this approach will help developers to spot more easily issues and artifacts in their NLG systems.

Future works could equip *LLM-as-a-qualitative-judge* with advanced reasoning or agentic pipelines, tune LLMs for issue report generation, and study the approach for a wider set of languages.

Limitations

As any LLM-based system, *LLM-as-a-qualitative-judge* can make occasional mistakes in analysis or clustering. In Section 5, we discuss types of such mistakes and recommend checking a couple of examples of each issue, which are also output by *LLM-as-a-qualitative-judge*.

Regarding limitations of the evaluation methodology, despite our efforts in considering a diverse set of tasks, domains, and LLMs, we acknowledge the infeasibility of covering the entire breadth of NLG applications and models in our study. Another limitation is that we mainly focus on English. We believe our findings will transfer to other languages, with the use of strong recent multilingual LLMs, but acknowledge that the reliability of *LLM-as-a-qualitative-judge* in multilingual studies requires a separate study.

Broader impact

We acknowledge that as any LLM-based system, *LLM-as-a-qualitative-judge* can make errors which could propagate to the downstream systems and decrease their performance. For example, if developers rely solely on the issue names formulated by Judge, this could occasionally lead to unnecessary or even harmful modifications of their NLG systems. This could also happen in case of misinterpetation of an issue by a developer due to issue subjectivity. To reduce such risks, we recommend developers to check examples of issue types, which are also output by *LLM-as-a-qualitative-judge*, in addition to the issue names and description.

Acknowledgments

We greatly appreciate the help of Alexandre Misrahi, Salah Aït-Mokhtar, and Maxime Louis. The project was initiated at the Advanced Language Processing School (ALPS 2025, https://alps.imag.fr).

A part of this work was carried out within the framework of the AugmentIA Chair, supported by the Fondation Grenoble INP thanks to the patronage of Artelia Group, and is affiliated with Laboratory of informatics in Grenoble (LIG). A part of this work received government funding managed by the French National Research Agency under France 2030, reference ANR-23-IACL-0006.

Tunde's research is conducted with the financial support of Research Ireland under Grant Number 12/RC/2289_P2 - Insight Research Ireland Centre for Data Analytics.

References

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, and 33 others. 2023. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805.

Konstantine Arkoudas, Nicolas Guenon des Mesnards, Melanie Rubino, Sandesh Swamy, Saarthak Khanna, Weiqi Sun, and Haidar Khan. 2022. PIZZA: A new benchmark for complex end-to-end task-oriented parsing. *CoRR*, abs/2212.00265.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, André F. T. Martins, Philipp Mondorf, Vera Neplenbroek, Sandro Pezzelle, Barbara Plank, David Schlangen, Alessandro Suglia, Aditya K Surikuchi, Ece Takmaz, and Alberto Testoni. 2024. Llms instead of human judges? a large scale empirical study across 20 nlp evaluation tasks. *Preprint*, arXiv:2406.18403.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. https://lmsys.org/blog/2023-03-30-vicuna/.

Elizabeth Clark, Shruti Rijhwani, Sebastian Gehrmann, Joshua Maynez, Roee Aharoni, Vitaly Nikolaev, Thibault Sellam, Aditya Siddhant, Dipanjan Das, and Ankur Parikh. 2023. SEAHORSE: A multilingual, multifaceted dataset for summarization evaluation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9397–9413, Singapore. Association for Computational Linguistics.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

Jonathan Cook, Tim Rocktäschel, Jakob Foerster, Dennis Aumiller, and Alex Wang. 2024. Ticking all the boxes: Generated checklists improve llm evaluation and generation. *Preprint*, arXiv:2410.03608.

John Dang, Shivalika Singh, Daniel D'souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, Sandra Kublik, Meor Amer, Viraat Aryabumi, Jon Ander Campos, Yi-Chern Tan, Tom Kocmi, Florian Strub, Nathan Grinsztajn, Yannis Flet-Berliac, and 26 others. 2024. Aya expanse: Combining research breakthroughs for a new multilingual frontier. *Preprint*, arXiv:2412.04261.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Daryna Dementieva, Daniil Moskovskiy, Nikolay Babakov, Abinew Ali Ayele, Naquee Rizwan, Florian Schneider, Xintong Wang, Seid Muhie Yimam, Dmitry Ustalov, Elisei Stakovskii, Alisa Smirnova, Ashraf Elnagar, Animesh Mukherjee, and Alexander Panchenko. 2024. Overview of the multilingual text detoxification task at PAN 2024. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2024), Grenoble, France, 9-12 September, 2024*, volume 3740 of *CEUR Workshop Proceedings*, pages 2432–2461. CEUR-WS.org.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Güney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *CoRR*, abs/1704.05179.

Nigel Fernandez, Alexander Scarlatos, and Andrew S. Lan. 2024. Syllabusqa: A course logistics question answering dataset. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 10344–10369. Association for Computational Linguistics.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2024. GPTScore: Evaluate as you desire. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6556–6576, Mexico City, Mexico. Association for Computational Linguistics.

Xinyang Geng, Arnav Gudibande, Hao Liu, Eric Wallace, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. Koala: A dialogue model for academic research. Blog post.

Aidan Gomez and Cohere for AI. 2024. Command r: Retrieval-augmented generation at production scale. `https://cohere.com/blog/command-r`.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Nuno M. Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André F. T. Martins. 2024. xCOMET: Transparent machine translation evaluation through fine-grained error detection. *Transactions of the Association for Computational Linguistics*, 12:979–995.

Rishav Hada, Varun Gumma, Mohamed Ahmed, Kalika Bali, and Sunayana Sitaram. 2024. METAL: Towards multilingual meta-evaluation. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2280–2298, Mexico City, Mexico. Association for Computational Linguistics.

Rujun Han, Peng Qi, Yuhao Zhang, Lan Liu, Juliette Burger, William Wang, Zhiheng Huang, Bing Xiang, and Dan Roth. 2023. Robustqa: Benchmarking the robustness of domain adaptation for open-domain question answering. In *ACL Findings 2023*.

Rujun Han, Yuhao Zhang, Peng Qi, Yumo Xu, Jenyuan Wang, Lan Liu, William Yang Wang, Bonan Min, and Vittorio Castelli. 2024. RAG-QA arena: Evaluating domain robustness for long-form retrieval augmented question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 4354–4374. Association for Computational Linguistics.

Chen Huang and Guoxiu He. 2024. Text clustering as classification with LLMs. *CoRR*, abs/2410.00927.

Dongfu Jiang, Yishan Li, Ge Zhang, Wenhao Huang, Bill Yuchen Lin, and Wenhu Chen. 2024. TIGERScore: Towards building explainable metric for all text generation tasks. *Transactions on Machine Learning Research*.

Zdeněk Kasner, Ondrej Platek, Patricia Schmidtova, Simone Balloccu, and Ondrej Dusek. 2024. factgenie: A framework for span-based evaluation of generated texts. In *Proceedings of the 17th International Natural Language Generation Conference: System Demonstrations*, pages 13–15, Tokyo, Japan. Association for Computational Linguistics.

Sanghoon Kim, Dahyun Kim, Chanjun Park, Wonsung Lee, Wonho Song, Yunsu Kim, Hyeonwoo Kim, Yungi Kim, Hyeonju Lee, Jihoo Kim, Changbae Ahn, Seonghoon Yang, Sukyung Lee, Hyunbyung Park, Gyoungjin Gim, Mikyoung Cha, Hwalsuk Lee, and Sunghun Kim. 2024a. SOLAR 10.7B: Scaling large language models with simple yet effective depth up-scaling. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pages 23–35, Mexico City, Mexico. Association for Computational Linguistics.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. 2024b. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations*.

Seungone Kim, Juyoung Suk, Ji Yong Cho, Shayne Longpre, Chaeeun Kim, Dongkeun Yoon, Guijin Son, Yejin Cho, Sheikh Shafayat, Jinheon Baek, Sue Hyun Park, Hyeonbin Hwang, Jinkyung Jo, Hyowon Cho, Haebin Shin, Seongyun Lee, Hanseok Oh, Noah Lee, Namgyu Ho, and 13 others. 2025. The BiGGen bench: A principled benchmark for fine-grained evaluation of language models with language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5877–5919, Albuquerque, New Mexico. Association for Computational Linguistics.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham

Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024c. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353, Miami, Florida, USA. Association for Computational Linguistics.

Tom Kocmi, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Rebecca Knowles, Philipp Koehn, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Michal Novák, Martin Popel, and Maja Popović. 2022. Findings of the 2022 conference on machine translation (WMT22). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 1–45, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 193–203, Tampere, Finland. European Association for Machine Translation.

Anastasia Krithara, Anastasios Nentidis, Konstantinos Bougiatiotis, and Georgios Paliouras. 2023. BioASQ-QA: A manually curated corpus for biomedical question answering. *Scientific Data*, 10(1):170.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, Lester James Validad Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2025. RewardBench: Evaluating reward models for language modeling. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1755–1797, Albuquerque, New Mexico. Association for Computational Linguistics.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D Manning, Christopher Re, Diana Acosta-Navas, Drew A. Hudson, and 31 others. 2023. Holistic evaluation of language models. *Transactions on Machine Learning Research*. Featured Certification, Expert Certification, Outstanding Certification.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Shayne Longpre, Yi Lu, and Joachim Daiber. 2021. MKQA: A linguistically diverse benchmark for multilingual open domain question answering. *Transactions of the Association for Computational Linguistics*, 9:1389–1406.

Justin K. Miller and Tristram J. Alexander. 2024. Human-interpretable clustering of short-text using large language models. *CoRR*, abs/2405.07278.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Alexandre Misrahi, Nadezhda Chirkova, Maxime Louis, and Vassilina Nikoulina. 2025. Adapting large language models for multi-domain retrieval-augmented-generation. *Preprint*, arXiv:2504.02411.

Mistral AI. 2024. Un ministral, des ministraux. Accessed May 19, 2025.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Stefano Perrella, Lorenzo Proietti, Alessandro Scirè, Niccolò Campolungo, and Roberto Navigli. 2022. MaTESe: Machine translation evaluation as a sequence tagging problem. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 569–577, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Alina Petukhova, João Pedro Matos-Carvalho, and Nuno Fachada. 2024. Text clustering with LLM embeddings. *CoRR*, abs/2403.15112.

José Pombal, Dongkeun Yoon, Patrick Fernandes, Ian Wu, Seungone Kim, Ricardo Rei, Graham Neubig, and André F. T. Martins. 2025. M-prometheus: A suite of open multilingual LLM judges. *CoRR*, abs/2504.04953.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Dongyu Ru, Lin Qiu, Xiangkun Hu, Tianhang Zhang, Peng Shi, Shuaichen Chang, Cheng Jiayang, Cunxiang Wang, Shichao Sun, Huanyu Li, Zizhao Zhang, Binjie Wang, Jiarong Jiang, Tong He, Zhiguo Wang, Pengfei Liu, Yue Zhang, and Zheng Zhang. 2024. Ragchecker: A fine-grained framework for diagnosing retrieval-augmented generation. In *Advances in Neural Information Processing Systems*, volume 37, pages 21999–22027. Curran Associates, Inc.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States. Association for Computational Linguistics.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Alon Talmor, Ori Yoran, Ronan Le Bras, Chandra Bhagavatula, Yoav Goldberg, Yejin Choi, and Jonathan Berant. 2021. Commonsenseqa 2.0: Exposing the limits of AI through gamification. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Thibaut Thonet, Laurent Besacier, and Jos Rozen. 2025. Elitr-bench: A meeting assistant benchmark for long-context language models. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 407–428. Association for Computational Linguistics.

Emiel van Miltenburg, Miruna Clinciu, Ondřej Dušek, Dimitra Gkatzia, Stephanie Inglis, Leo Leppänen, Saad Mahamood, Emma Manning, Stephanie Schoch, Craig Thomson, and Luou Wen. 2021. Underreporting of errors in NLG output, and what to do about it. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 140–153, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Emiel van Miltenburg, Miruna Clinciu, Ondřej Dušek, Dimitra Gkatzia, Stephanie Inglis, Leo Leppänen, Saad Mahamood, Stephanie Schoch, Craig Thomson, and Luou Wen. 2023. Barriers and enabling factors for error analysis in NLG research. *Northern European Journal of Language Technology*, 9.

Vijay Viswanathan, Kiril Gashteovski, Carolin Lawrence, Tongshuang Wu, and Graham Neubig. 2024. Large language models enable few-shot clustering. *Trans. Assoc. Comput. Linguistics*, 12:321–333.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*.

Wenda Xu, Danqing Wang, Liangming Pan, Zhenqiao Song, Markus Freitag, William Wang, and Lei Li. 2023. INSTRUCTSCORE: Towards explainable text generation evaluation with automatic feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5967–5994, Singapore. Association for Computational Linguistics.

Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. 2024. FLASK: Fine-grained language model evaluation based on alignment skill sets. In *The Twelfth International Conference on Learning Representations*.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. 2024. Evaluating large language models at evaluating instruction following. In *The Twelfth International Conference on Learning Representations*.

Yuwei Zhang, Zihan Wang, and Jingbo Shang. 2023. ClusterLLM: Large language models as a guide for text clustering. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 13903–13920. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023a. Judging llm-as-a-judge with mt-bench and chatbot arena. volume 36, pages 46595–46623.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023b. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

A Details on data annotation

Per-instance analysis. The core of our meta-evaluation strategy is to collect manual annotations of failure cases for a set of instances from various tasks and domains. For each instance, consisting of a task input, a ground truth response, a generated response, a description of a task metric, and optionally retrieved documents, an annotator's task is to formulate what the particular issue is in this instance. We then prompt an *evaluator LLM* to judge whether the issue explanation determined by the *LLM-as-a-qualitative-judge* for a particular instance matches the issue determined by the human annotator. The outputs from the *evaluator LLM* are binary and can be accumulated into a *per-instance analysis accuracy* score.

The annotation instruction asks to ignore instances which have multiple issues (to avoid ambiguity in per-instance analysis), instances where ground truth labels appear to be wrong, and instances where the annotator's expertise is insufficient to judge the correctness of the generated answer. We also limit the number of instances with the same issue to not exceed 8 examples per dataset, to ensure the diversity of the final dataset.

Issue clustering. Human annotation also includes a step of manually clustering issues discovered in per-instance analysis, i.e., specifying cluster indices and cluster names (generalized issue types) for labeled instances. This annotation is then used to compute clustering agreement between the clustering produced by *LLM-as-a-qualitative-judge* and by a human annotator.

Dataset composition. For each considered dataset, we manually label failures in up to 40 generations from one of the open-source LLMs (`Qwen-2.5-1b`, `Llama-3.2-1b`, `Command-R-35b`, `Vicuna-1.5-13B`).

Annotation details. All the labeling was performed by the authors of the paper in Google Spreadsheets[4]. Each instance was annotated by one author. Authors of the paper are PhD students in the NLP field or have already completed their PhD in NLP and are employed as NLP researchers.

Time needed for data annotation varies between tasks: it took us 1—6 hours per task. Tasks with longer inputs, e.g. RA-QA, and from more complex domains, e.g. biomedical, take more time to annotate, e.g. they require reading the retrieved documents carefully.

Annotation instruction is provided in Figure 6.

Inter-annotator agreement. We measure the inter-annotator agreement on a subset of 100 instances, i.e. each of these instances was labeled by two annotators and then we computed their agreement using the same evaluator LLM as in other experiments, i.e. Claude-3.7-sonnet. The resulting inter-annotator agreement was 57% (percentage of cases when two annotators suggested the same issue, as judged by Claude-3.7-sonnet), i.e. the similar range as the scores we obtain in Table 2.

The main factor contributing to the moderate agreement is the subjectivity of issue analysis. For example, in a situation when generation was stopped due to reaching the maximum new tokens limit, one annotator said "The response is incomplete" and another annotator said "Generation was stopped too early". Both denote the same root issue, but are formulated differently and Claude judges these comments as different.

B Further details on the experimental setup

Case study. For each of the three considered BigBenchHars tasks, we build a simple initial generative pipeline. This pipeline is then improved in two rounds by generating issue reports with *LLM-as-a-qualitative-judge*. Configurations of the initial pipeline are as follows. System prompt: "*You are a helpful assistant. Output your answer after a final separator 'Answer:'*". Generation hyperparameters: all hyperparameters set to default values from the HuggingFace or OpenAI API, plus setting maximum new tokens or 500 for HuggingFace models. The final answers are obtained by cropping the content after a final separator "*Answer:*" and applying a `.strip()` python function. Evaluation function: exact match with ground truth. *LLM-as-a-qualitative-judge* is run with GPT-4.1 and providing a one-sentence description of a task metric, i.e. "Evaluation is conducted using exact matching between the ground-truth label and the content of the generated response after the final separator 'Answer:'".

Meta-evaluation experiments. For each instance, *LLM-as-a-qualitative-judge* is provided with a task input, a ground truth response, a generated response, 5 retrieved documents (only for

[4] `https://docs.google.com/spreadsheets`

> For each example, consisting of a user prompt, a ground truth label, an LLM generation, and optionally retrieved documents, an annotator's task is to formulate what is a particular failure case in this example. Identify only one, most important specific, clearly visible issue in each test case. Please formulate the detected issue as a clear, full sentence, e.g. "The generated response is in German instead of French which is the language of the user input" or "The retrieved documents are from a datastore which is irrelevant to the given user question".
> Please add your annotations in the following Google Spreadsheet: [link], column “Per-instance analysis”. You can skip instances (rows) for which you feel that you do not have enough expertise to detect an issue, which have multiple issues, or for which ground truth labels appear to be wrong.
> After annotating per-instance analysis, suggest a clustering of the detected issues, i.e. how would you group them, and add the corresponding cluster indexes and names in columns “Cluster index” and “Cluster name”.

Figure 6: Annotation instruction for meta-evaluation data.

RA-QA), and a short task comment. The task comment describes the task metric (in one sentence), provides a comment on the nature of ground truth responses (either that it is the expected answer or that it is only one of the possible correct answers), and also contains a comment that retrieval-augmented generation (RAG) or Chain-of-Thought (COT) prompting was used, when applicable (6 datasets with RAG and 2 datasets with COT). The used task metrics are binary LLM-as-a-judge (the generated response is accepted or not) or binary Match (outputs `True` is one of the ground truth answers is contained a substring in the generated response, and `False` otherwise). Task comments for all datasets are also presented in Appendix H.

For *LLM-as-a-qualitative-judge*, open-source LLMs are run on a single V100 GPU with greedy decoding (∼20 GPU-hours in total). Commercial LLMs are run via API with requesting `json` output format.

The time of running the *LLM-as-a-qualitative-judge* algorithm depends on the setting (commercial vs open-source LLMs, type of GPU etc) and in our experiments was taking 2—30 min, i.e. reasonably short on the scale of the time needed to develop an NLG system.

Meta-evaluation of an evaluator LLM. To ensure the reliability of the *evaluator LLM*, we collected a small meta-evaluation dataset of 50 instances from 4 datasets (MKQA (ru), RobustQA Writing, FLASK, MultiDetox), where the equivalence of the *LLM-as-a-qualitative-judge*'s and *human annotator's* per-instance analysis was judged by a human annotator and can be compared to the *evaluator LLM*'s verdict. Strong commercial LLMs, such as `GPT-4o`, `Gemini-2.0-Flash`, and `Claude-3.7-Sonnet`, achieved a meta-evaluation accuracy of 85-90% on this dataset, and an open-source `Solar-10.7B` (Kim et al., 2024a) achieved a meta-accuracy of 60%. In all the experiments, we use `claude-3-7-sonnet-20250219` as the *evaluator LLM*, to avoid using the same LLM for analysis and for evaluation.

C Clustering experiment setup

In this experiment, we perform a hyperparameter grid search for five clustering algorithms: KMeans, Agglomerative Clustering, Spectral Clustering, Gaussian Mixture Models (GMM), and HDBSCAN on a synthetic set. Each algorithm is evaluated across a range of hyperparameter combinations. For KMeans, we vary the `distance_metric` (euclidean, cosine), `kmeans_init` strategy (k-means++, random), `kmeans_n_init` (10, 50), and `kmeans_max_iter` (300, 500). For Agglomerative Clustering, we test all combinations of `distance_metric` (euclidean, cosine) and `linkage_type` (ward, average, complete), while ensuring that ward is only paired with euclidean (as required by the algorithm). Spectral Clustering configurations include `distance_metric` (euclidean, cosine), `assign_labels` (kmeans, discretize), `spectral_gamma` (0.1, 0.5, 1.0, 2.0), and spectral_n_neighbors (5, 10, 20). For GMM, we explore covariance_type (full, diag), gmm_init_params (kmeans, random), and gmm_max_iter (100, 300). Lastly, HDBSCAN is tested with distance_metric (euclidean, cosine), min_cluster_size (3, 5, 10, 15, 20), hdbscan_min_samples (None, 1, 5), and hdbscan_cluster_selection_method (eom, leaf). Each valid configuration is evaluated over three independent trials with different random seeds to ensure robustness. After collecting results based on Adjusted Rand Index (ARI), the best-performing configuration for each algorithm on the synthetic validation set is selected. These best configurations are then applied to the test set of synthetic data and to the real dataset.

D Models

Model	BibTeX	License	Model Repository
GPT-4o	(OpenAI et al., 2024)	Proprietary	`https://platform.openai.com/docs/models/gpt-4o`
Gemini-2.0-Flash	(Anil et al., 2023)	Proprietary	`https://deepmind.google/technologies/gemini/flash/`
Qwen-2.5	(Qwen et al., 2025)	Apache 2.0	`https://huggingface.co/collections/Qwen/qwen25-66e81a666513e518adb90d9e`
DeepSeek-R1-Distill-Llama-8B	(DeepSeek-AI, 2025)	Llama 3.1 Community License	https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Llama-8B
Aya-Expanse-8B	(Dang et al., 2024)	Creative Commons Attribution Non Commercial 4.0	https://huggingface.co/CohereLabs/aya-expanse-8b
Llama-3.1-8B-Instruct	(Grattafiori et al., 2024)	Llama 3.1 Community License	https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct
Llama-3.2-1B-Instruct	(Grattafiori et al., 2024)	Llama 3.2 Community License	https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct
Ministral-8B-Instruct	(Mistral AI, 2024)	Mistral AI Research License	https://huggingface.co/mistralai/Ministral-8B-Instruct-2410
Solar-10.7B	(Kim et al., 2024a)	Creative Commons Attribution Non Commercial 4.0	https://huggingface.co/upstage/SOLAR-10.7B-Instruct-v1.0
Vicuna-1.5-13B	(Chiang et al., 2023)	Llama 2 Community License Agreement	`https://huggingface.co/lmsys/vicuna-13b-v1.5`
Command-R-35B	(Gomez and for AI, 2024)	Creative Commons Attribution Non Commercial 4.0	`https://huggingface.co/CohereLabs/c4ai-command-r-v01`

Table 4: References to the used LLMs; all LLMs allow use for research.

E Datasets

Dataset name	Dataset reference	License
Natural Language Generation		
FLASK data mix: Self-Instruct, WizardLM, Koala, CommonSense QA	(Wang et al., 2023; Xu et al., 2024; Geng et al., 2023; Talmor et al., 2021)	Apache 2.0, MIT, Apache 2.0, Creative Commons Attribution 4.0
WMT'22	(Kocmi et al., 2022)	Apache 2.0
Elitr-Bench	(Thonet et al., 2025)	Attribution 4.0 International
PIZZA	(Arkoudas et al., 2022)	Attribution-NonCommercial 4.0 International
GSM8K	(Cobbe et al., 2021)	MIT License
ParaDetox	(Dementieva et al., 2024)	OpenRAIL++
Retrieval-augmented QA		
MKQA (ru)	(Longpre et al., 2021)	Creative Commons Attribution-ShareAlike 3.0 Unported License
BioASQ	(Krithara et al., 2023)	Attribution 2.5 Generic
RobustQA	(Santhanam et al., 2022; Han et al., 2024, 2023)	Apache-2.0
SearchQA	(Dunn et al., 2017)	BSD 3-Clause
SyllabusQA	(Fernandez et al., 2024)	Attribution-NonCommercialShareAlike
BigBenchHard		
Date understanding; Word sorting; Movie recommendation	(Suzgun et al., 2022)	MIT

Table 5: References to the used datasets; all datasets allow use for research. We select instances from test splits.

F Per-dataset results

Table 6 presents per-dataset results for for GPT-4o as *LLM-as-a-qualitative-judge*.

G Prompts

Figures 8–11 present prompts used for per-instance analysis, issue clustering, and evaluation.

Dataset	Per-inst. an. acc. (%)	Issue clust. ARI	Issue clust. SLC
Semantic parsing	94.1	0.41	0.29
Grade school math	88.2	0.04	0.22
Detoxification	77.8	0.36	0.28
Long-context QA	69.2	0.07	0.50
Translation en-ru	65.8	0.10	0.63
Instruction following	55.9	0.09	0.19
RA-QA: SyllabusQA	77.8	0.16	0.55
RA-QA: MKQA (ru)	75.8	0.17	0.44
RA-QA: BioASQ	66.7	0.08	0.31
RA-QA: SearchQA	38.4	0.15	0.16
RA-QA: Writing	30.7	0.00	0.11
RA-QA: Lifestyle	19.0	0.00	0.32

Table 6: Per-dataset results for GPT-4o as *LLM-as-a-qualitative-judge*.

```
You are an expert in analysing the failure cases in natural language generation tasks
You are given a question, ground truth label(s), and the answer generated by an LLM.
The generated answer was not accepted by the automatic evaluation measured with metric {metric info}
Read all these materials and reply what is the particular failure case in this example, i. e. why exactly the generated response was not accepted.
The problem can be in any part of the pipeline, including the question itself or any aspects of the system outputs.
IMPORTANT: identify ONE, MOST IMPORTANT, SPECIFIC, CLEARLY VISIBLE issue in each test case.

Question:
***
{question}
***

Ground truth label(s):
***
{label}
***

LLM-generated answer:
***
{answer}
***

So what is the particular failure in this example? First output a detailed analysis, and then output a final summary of the failure in one or two sentences after a special separator “Summary:”.
```

Figure 7: Prompt used for per-instance analysis. The presented version of the prompt is for text LLM outputs, the prompt can be easily changed if JSON outputs are supported by an LLM. For RA-QA, we also include retrieved documents in the prompt.

```
You are an expert in analysing the failure cases of natural language generation systems.
You already performed per-example analysis, where for each example, you were given a question, ground truth label(s), and the answer generated by an LLM.
The generated answer in all examples was not accepted by the automatic evaluation.
You already read all these materials and formulated what was the particular failure case in each example, i.e. which part of the pipeline failed so that the generated response was not accepted.

Now your task is to summarize all your per-example analyses into a concise overall summary of failure cases for the given dataset.
Summarize what are the various failure types in this dataset (provide the overall count of each error type and also ids of all examples of each error type).
Please try to be very specific in determining error types, avoid too much generic error types.  On the contrary, determine as much as possible FINE GRAINED error types.
Furthermore, provide a comment for each error type explaining the essence of this error type in a bit more details (in the context of this particular dataset).

*** Per-example analysis which you generated before ***
{analysis}
*** per-example analysis ended ***

Summarize all your per-example analyses into a concise overall summary of failure cases.
Generate a json with the only key “summary”, and a value is a dict of error types. Each value in this dict (corresponding to one detected error type) is a dictionary with keys “error_name”, “error_description”, “indexes” (indexes of all examples with this error type), and “num_examples” (overall count of this error type).
```

Figure 8: Prompt used for issue clustering with direct prompting.

```
You are an expert in analysing the failure cases of natural language generation systems.
You already performed per-example analysis, where for each example, you were given a question, ground truth label(s), and the answer generated by an LLM.
The generated answer in all examples was not accepted by the automatic evaluation.
You already read all these materials and formulated what was the particular failure case in each example, i.e. which part of the pipeline failed so that the generated response was not accepted.


Now your task is to summarize all your per-example analyses into a concise overall summary of failure cases for the given dataset.
You are using a SPECIAL CUMULATIVE ALGORITHM as follows. You are going through examples one by one and accumulate discovered error cases in a special pool. For each example, you check if any of the already discovered error types from the pool fits it, and if so, you assign this error type to this example. If none of already existing error types fit the current example, you create a new error type and add it to a pool.


A pool of already discovered error cases:
***
{error cases}
***


Analysis of a current example:
***
{analysis}
***


Do you think any of the already discovered error cases fit the current example?
If yes, output “Decision:” and a key marking the chosen error case, e.g. “Decision: type_0”. Do it only if you are REALLY sure that the chosen error case fits the current example! DO NOT output cluster name. If not, output “Decision: None”. DO NOT output anything else.
```

Figure 9: Prompt used for classifying each instance in the cumulative clustering strategy. The presented version of the prompt is for text LLM outputs, the prompt can be easily changed if JSON outputs are supported by an LLM.

```
You are an expert in analysing the failure cases of natural language generation
systems.
You already performed per-example analysis, where for each example, you were
given a question, ground truth label(s), and the answer generated by an LLM.
The generated answer in all examples was not accepted by the automatic
evaluation.
You already read all these materials and formulated what was the particular
failure case in each example, i.e. which part of the pipeline failed so that
the generated response was not accepted.
Now your task is to summarize all your per-example analyses into a concise
overall summary of failure cases for the given dataset.
You are using a SPECIAL CUMULATIVE ALGORITHM as follows. You are going through
examples one by one and accumulate discovered error cases in a special pool.
For each example, you check if any of the already discovered error types from
the pool fits it, and if so, you assign this error type to this example. If
none of already existing error types fit the current example, you create a new
error type and add it to a pool.


A pool of already discovered error cases:
***
{error cases}
***


Analysis of a current example:
***
{analysis}
***


You decided to create a new error type for a given example, not yet present in
a pool. Now you need to generate SHORT LABEL and a 1 or 2 SENTENCE DESCRIPTION
for this new error type. Please try to be very specific in determining a FINE
GRAINED error type, avoid too much generic error types. At the same time, it
is important that the generated label and description of the error type can
be GENERALIZED to other examples, i.e. avoid references to the particular
content of the current example (names, dates, etc): anything related ONLY to
this example SHOULD NOT be present in the description and label.
Output answer is the following format: ”SHORT LABEL: DESCRIPTION”, do not output
anything else!
```

Figure 10: Prompt used for generating a new issue type in the cumulative clustering strategy. The presented version of the prompt is for text LLM outputs, the prompt can be easily changed if JSON outputs are supported by an LLM.

```
Situation: Two experts are inspecting examples in natural language generation
for a particular dataset.
You will be given 2 sentences, which represent the conclusions of the two
experts about the same example, i.e. what is a failure case in the example
they were given.
Your task is to determine if the experts describe the same failure.


Expert 1 conclusion: E1
Expert 2 conclusion: E2


Do the experts describe the same failure? Output only one word ‘Yes’ or ‘No’.
```

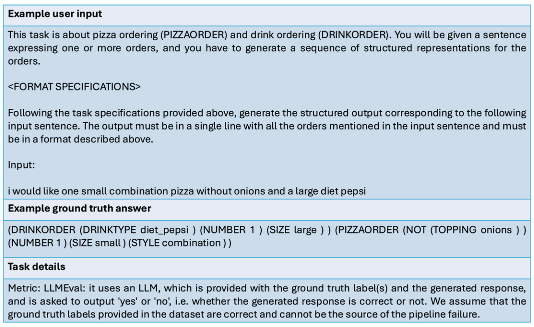
Figure 11: Prompt used for evaluation. The presented version of the prompt is for text LLM outputs, the prompt can be easily changed if JSON outputs are supported by an LLM.

H Examples

The following pages present examples of task instances, per-instance analysis, generated error reports, and clustering confusion matrices, for all 12 considered datasets. Clusters of size 1 are shown in confusion matrices but omitted in error reports, for space purposes.

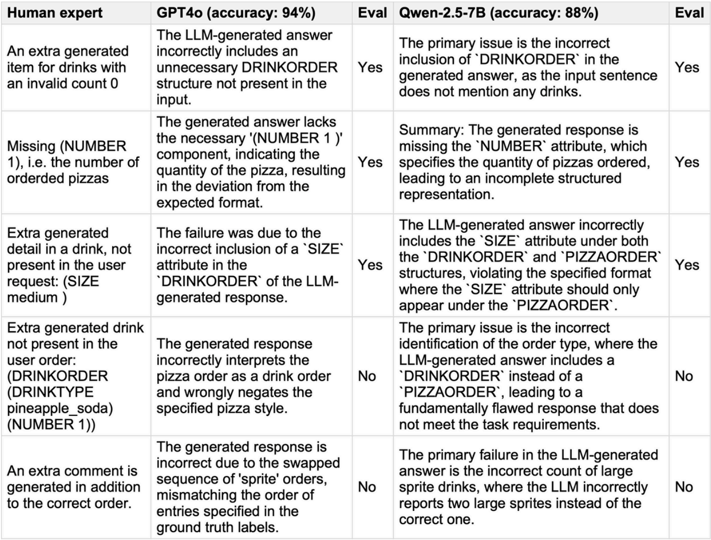
H.1 Semantic parsing (PIZZA dataset)

Task illustration



Example user input
This task is about pizza ordering (PIZZAORDER) and drink ordering (DRINKORDER). You will be given a sentence expressing one or more orders, and you have to generate a sequence of structured representations for the orders. <FORMAT SPECIFICATIONS> Following the task specifications provided above, generate the structured output corresponding to the following input sentence. The output must be in a single line with all the orders mentioned in the input sentence and must be in a format described above. Input: i would like one small combination pizza without onions and a large diet pepsi
Example ground truth answer
(DRINKORDER (DRINKTYPE diet_pepsi) (NUMBER 1) (SIZE large)) (PIZZAORDER (NOT (TOPPING onions)) (NUMBER 1) (SIZE small) (STYLE combination))
Task details
Metric: LLMEval: it uses an LLM, which is provided with the ground truth label(s) and the generated response, and is asked to output 'yes' or 'no', i.e. whether the generated response is correct or not. We assume that the ground truth labels provided in the dataset are correct and cannot be the source of the pipeline failure.

Examples of per-instance analysis illustration



Human expert	GPT4o (accuracy: 94%)	Eval	Qwen-2.5-7B (accuracy: 88%)	Eval
An extra generated item for drinks with an invalid count 0	The LLM-generated answer incorrectly includes an unnecessary DRINKORDER structure not present in the input.	Yes	The primary issue is the incorrect inclusion of `DRINKORDER` in the generated answer, as the input sentence does not mention any drinks.	Yes
Missing (NUMBER 1), i.e. the number of orderded pizzas	The generated answer lacks the necessary '(NUMBER 1)' component, indicating the quantity of the pizza, resulting in the deviation from the expected format.	Yes	Summary: The generated response is missing the `NUMBER` attribute, which specifies the quantity of pizzas ordered, leading to an incomplete structured representation.	Yes
Extra generated detail in a drink, not present in the user request: (SIZE medium)	The failure was due to the incorrect inclusion of a `SIZE` attribute in the `DRINKORDER` of the LLM-generated response.	Yes	The LLM-generated answer incorrectly includes the `SIZE` attribute under both the `DRINKORDER` and `PIZZAORDER` structures, violating the specified format where the `SIZE` attribute should only appear under the `PIZZAORDER`.	Yes
Extra generated drink not present in the user order: (DRINKORDER (DRINKTYPE pineapple_soda) (NUMBER 1))	The generated response incorrectly interprets the pizza order as a drink order and wrongly negates the specified pizza style.	No	The primary issue is the incorrect identification of the order type, where the LLM-generated answer includes a `DRINKORDER` instead of a `PIZZAORDER`, leading to a fundamentally flawed response that does not meet the task requirements.	No
An extra comment is generated in addition to the correct order.	The generated response is incorrect due to the swapped sequence of 'sprite' orders, mismatching the order of entries specified in the ground truth labels.	No	The primary failure in the LLM-generated answer is the incorrect count of large sprite drinks, where the LLM incorrectly reports two large sprites instead of the correct one.	No

An example of a final report generated by LLM-as-a-qualitative-judge (GPT4o)

* **Entity Mislabeling:** The generated answer contains incorrect labeling of entities, leading to a mismatch with the expected ground truth label. (12)

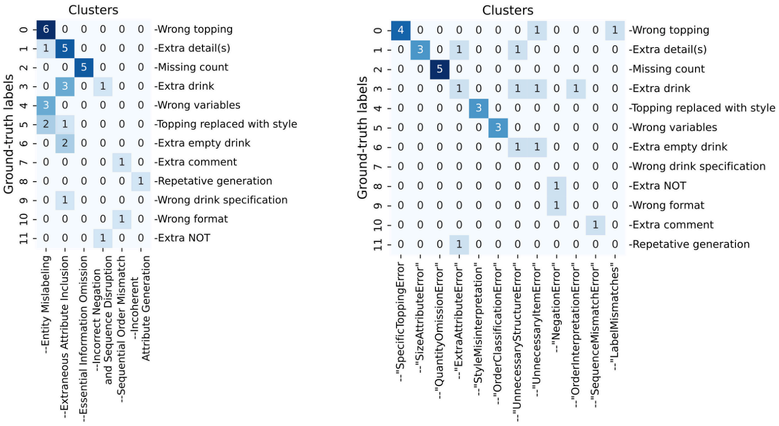
* **Extraneous Attribute Inclusion:** The generated answer includes attributes that are not part of the ground truth requirements, leading to discrepancies in the expected output. (12)

* **Essential Information Omission:** The generated response fails to include critical quantifiable details necessary for completing a structured output. (5)

* **Sequential Order Mismatch:** The generated response fails to match the required sequence of elements as specified in the ground truth, leading to erroneous outputs. (2)

* **Incorrect Negation and Sequence Disruption:** The generated response incorrectly negates an intended attribute and alters the sequence of elements, resulting in a deviation from the expected output. (2)

Confusion matrices for issue type clustering for GPT4o (left) and Qwen-2.5-7B (right)



H.2 Long context QA (Elitr-Bench dataset)

Task illustration

Example user input

You are given a meeting transcript. You task is to answer a question given below about this meeting. Do not output anything else.

<TRANSCRIPT>

Question:

Who is interested in applying to a PhD?

Example ground truth answer

An Indian student known by [PERSON18]

Task details

Metric: LLMEval: it uses an LLM, which is provided with the ground truth label(s) and the generated response, and is asked to output 'yes' or 'no', i.e. whether the generated response is correct or not. We assume that the ground truth labels provided in the dataset are correct and cannot be the source of the pipeline failure.

Examples of per-instance analysis illustration

Human expert	GPT4o (accuracy: 69%)	Eval	Qwen-2.5-7B (accuracy: 65%)	Eval
An LLM outputs a partially correct list of persons.	The generated response failed to mention [PERSON15] as involved in the specific call, as indicated in the ground truth labels.	Yes	The generated response fails to identify a specific person who will have a call to discuss multi-source experiments, instead providing details about the sequence of discussions and tasks.	No
Wrogn answer: end of March instead of end of February.	The LLM failed to extract and correctly interpret the timeline for de-identification from the meeting transcript, leading to an incorrect response.	Yes	Summary: The generated response incorrectly specifies the completion date as the end of March instead of the end of February as per the ground truth label.	Yes
An LLM provides a generic explanation of a term "ladder climbing" instead of providing an explanation contextualized in the discussion, i.e. STT-specific process with HTML tables	The generated answer fails to address the specific context of 'ladder climbing' as discussed in the meeting, focusing instead on a generic definition not tailored to the transcript.	Yes	The generated answer failed to correctly interpret "ladder climbing" in the context of the meeting transcript, instead providing a general explanation of model improvement techniques unrelated to the specific STT system evaluation process mentioned in the ground truth label.	Yes
An LLM mistakenly claims that the information provided in the context is insufficient to answer the given question.	The LLM-generated answer failed to extract and correctly report the specific number of people mentioned in the program committee from the transcript.	No	Summary: The LLM failed to extract and provide the specific number of people in the program committee as stated in the ground truth label.	No
The generated response does not answer the given question and talks about a different topic.	The LLM-generated answer failed to identify and cite the geographical locations of the meeting participants as required by the question.	No	The generated response failed to identify places from which participants spoke in the meeting, instead providing a summary of the discussion topics and participant actions.	No

An example of a final report generated by LLM-as-a-qualitative-judge (GPT4o)

* **Entity Misidentification:** The generated response incorrectly identifies or attributes an action to an entity, leading to a mismatch with the expected answer. (6)

* **Incomplete Entity Mention:** The generated response fails to include all entities required by the context, resulting in an incomplete answer. (5)

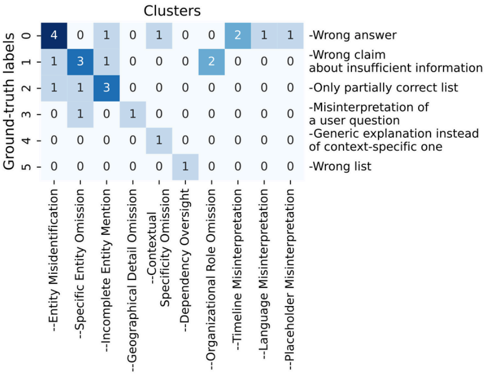
* **Specific Entity Omission:** The generated response fails to address the request for information about a specific entity, resulting in an incomplete answer that lacks relevant details. (5)

* **Organizational Role Omission:** The generated response fails to accurately acknowledge the participation or role of entities in a collaborative activity, resulting in an evasive or incomplete answer. (2)

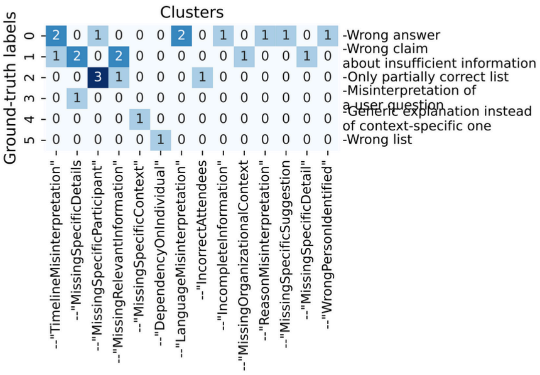
* **Contextual Specificity Omission:** The generated response fails to integrate specific contextual details relevant to a particular discussion, resulting in an overly generic answer that does not align with the nuanced requirements of the query. (2)

* **Timeline Misinterpretation:** The generated response fails to accurately identify and interpret chronological information from the given context, resulting in an incorrect timeline being communicated. (2)

Confusion matrices for issue type clustering for GPT4o (left) and Qwen-2.5-7B (right)



Ground-truth labels	--Entity Misidentification	--Specific Entity Omission	--Incomplete Entity Mention	--Geographical Detail Omission	--Contextual Specificity Omission	--Dependency Oversight	--Organizational Role Omission	--Timeline Misinterpretation	--Language Misinterpretation	--Placeholder Misinterpretation	
0	4	0	1	0	1	0	0	2	1	1	-Wrong answer
1	1	3	1	0	0	0	2	0	0	0	-Wrong claim about insufficient information
2	1	1	3	0	0	0	0	0	0	0	-Only partially correct list
3	0	1	0	1	0	0	0	0	0	0	-Misinterpretation of a user question
4	0	0	0	0	1	0	0	0	0	0	-Generic explanation instead of context-specific one
5	0	0	0	0	0	1	0	0	0	0	-Wrong list



Ground-truth labels	--"TimelineMisinterpretation"	--"MissingSpecificDetails	--"MissingSpecificParticipant"	--"MissingRelevantInformation"	--"MissingSpecificContext"	--"DependencyOnIndividual"	--"LanguageMisinterpretation"	--"IncorrectAttendees	--"IncompleteInformation"	--"MissingOrganizationalContext	--"ReasonMisinterpretation"	--"MissingSpecificSuggestion	--"MissingSpecificDetail"	--"WrongPersonIdentified"	
0	2	0	1	0	0	0	2	0	1	0	1	1	0	1	-Wrong answer
1	1	2	0	2	0	0	0	0	0	1	0	0	1	0	-Wrong claim about insufficient information
2	0	0	3	1	0	0	0	1	0	0	0	0	0	0	-Only partially correct list
3	0	1	0	0	0	0	0	0	0	0	0	0	0	0	-Misinterpretation of a user question
4	0	0	0	0	1	0	0	0	0	0	0	0	0	0	-Generic explanation instead of context-specific one
5	0	0	0	0	0	1	0	0	0	0	0	0	0	0	-Wrong list

H.3 Detoxification (MultiDetox dataset)

Task illustration

Example user input
You are a professional moderator of content on social networks. You will be given an input sentence and your task is to output a refined sentence, which satisfies the following conditions: * if the given sentence is not toxic, then copy the input sentence * if the given sentence is toxic, then output a new sentence with the meaning as close as possible to the given sentence, in the same language as the given sentence. Given sentence: {toxic sentence} Output a refined sentence. Only output one refined sentence, do not output anything else.
Example ground truth answer
{detoxified sentence}
Task details
Metric: LLMEval: it uses an LLM, which is provided with the ground truth label(s) and the generated response, and is asked to output 'yes' or 'no', i.e. whether the generated response is correct or not. While ground truth labels provide correct high-quality answers, it does NOT mean that an LLM has to guess 100% all the information from the ground truth; LLM also may provide a valid alternative answer.

Examples of per-instance analysis illustration

Human expert	GPT4o (accuracy: 78%)	Eval	Qwen-2.5-7B (accuracy: 61%)	Eval
The generated sentence is in English instead of Russian.	The generated response failed by translating the Russian input into English, violating the requirement for language consistency.	Yes	The primary failure is the language mismatch between the input and output sentences, leading to a significant deviation from the required criteria.	Yes
A part of an input sentence was ommited, i.e. "Kid"	The LLM-generated response was not accepted because it omitted the word 'kid,' altering the tone of direct address in the sentence.	Yes	The failure lies in the loss of the specific reference to "kid" in the refined sentence, which changes the intended meaning and context of the original toxic statement.	Yes
The sentence was not fully detoxified.	The generated sentence retains offensive language from the input, failing to fully refine it into a less toxic version as required.	Yes	The primary issue is that the generated response softens the tone of the original toxic sentence, failing to preserve its intended harshness and directness.	No
An LLM refuses to answer the user's request due to its safety guideline.	The LLM response did not refine the toxic input sentence as required by the task.	No	The primary failure is that the generated answer does not provide a non-toxic sentence with a similar meaning to the input, instead offering a generic and irrelevant response.	No
Model did not follow the instructions to only output the refined sentence, i.e. it also output a comment.	The generated response failed because it added assumptions about the tone and content not present in the original sentence, violating the requirement to maintain the original meaning closely.	No	The failure lies in the introduction of an unnecessary opinion and the addition of subjective language, which deviates from the requirement to maintain the exact meaning of the original sentence in a more refined manner without altering its core message.	Yes

An example of a final report generated by LLM-as-a-qualitative-judge (GPT4o)

* **Inadequate Toxicity Mitigation:** The system fails to reduce or neutralize the presence of toxic sentiment in the generated response while maintaining the original intended meaning. (12)

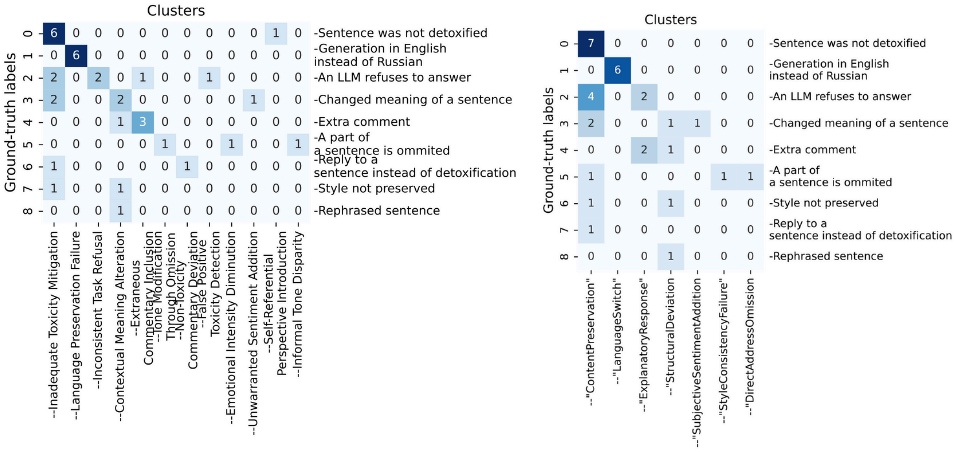
* **Language Preservation Failure:** The system fails to maintain the original language of the input when generating a response, resulting in an unwanted translation or language shift. (6)

* **Contextual Meaning Alteration:** The system significantly changes the original meaning of the input while attempting to refine it, resulting in a response that deviates from the intended context. (5)

* **Extraneous Commentary Inclusion:** The system introduces additional commentary or information that was not requested, instead of providing a direct and concise response to the task. (4)

* **Inconsistent Task Refusal:** The system declines to perform a task based on content sensitivity without explicitly following the task instructions, resulting in a failure to generate the required output. (2)

Confusion matrices for issue type clustering for GPT4o (left) and Qwen-2.5-7B (right)



H.4 Translation en-ru (WMT’22 dataset)

Task illustration

Example user input

Translate a sentence from English to Russian, do not output anything else:

I can see here that there is no rider assigned yet to your order.

Example ground truth answer

Я вижу, что на ваш заказ еще не назначен водитель.

Task details

Metric: LLMEval: it uses an LLM, which is provided with the ground truth label(s) and the generated response, and is asked to output 'yes' or 'no', i.e. whether the generated response is correct or not. While ground truth labels provide correct high-quality answers, it does NOT mean that an LLM has to guess 100% all the information from the ground truth; LLM also may provide a valid alternative answer.

Examples of per-instance analysis illustration

Human expert	GPT4o (accuracy: 66%)	Eval	Qwen-2.5-7B (accuracy: 71%)	Eval
Mistranslation: the wrong word is used to translate the word "rider".	The error was due to the LLM's mistranslation of 'rider', using a term more aligned with racing rather than the intended meaning of 'driver' for transportation services.	Yes	The primary issue lies in the translation of "rider" as "гонщиком" instead of "водителем", which is less precise and potentially misleading in the context of the original sentence.	Yes
Mistranslation: the wrong word is used to translate the word "generic".	The LLM-generated translation uses a less common equivalent for 'generic' and a less standard term for 'cord', reducing clarity.	Yes	The primary failure in the generated response is the incorrect translation of "generic" as "родовой" instead of "универсальный."	Yes
Inappropriate terminology: the word "rider" is transliterated instead of being translated into a proper Russian term	The LLM incorrectly translated 'rider' as 'райдер' instead of 'водитель', leading to an inaccurate translation.	Yes	The primary issue is the incorrect use of "райдер" instead of "водитель" in the translation, which slightly deviates from the ground truth label and reduces the accuracy of the response.	Yes
One input sentence is split into two output sentences	The LLM mistranslated a key phrase, altering the meaning of the sentence.	No	Summary: The LLM failed to correctly translate the phrase "once the order has been placed" to "после того, как заказ был размещен," resulting in a significant change in the meaning of the sentence.	No
Two words similar in meaning, Repair/restore, are translated into the same word in Russian.	The failure is due to incorrect capitalization and slight mistranslation in the LLM-generated response.	No	The primary issue is the incorrect placement of the verb "коснитесь" (tap/touch) between the two phrases, which disrupts the intended meaning and flow of the command.	No

An example of a final report generated by LLM-as-a-qualitative-judge (GPT4o)

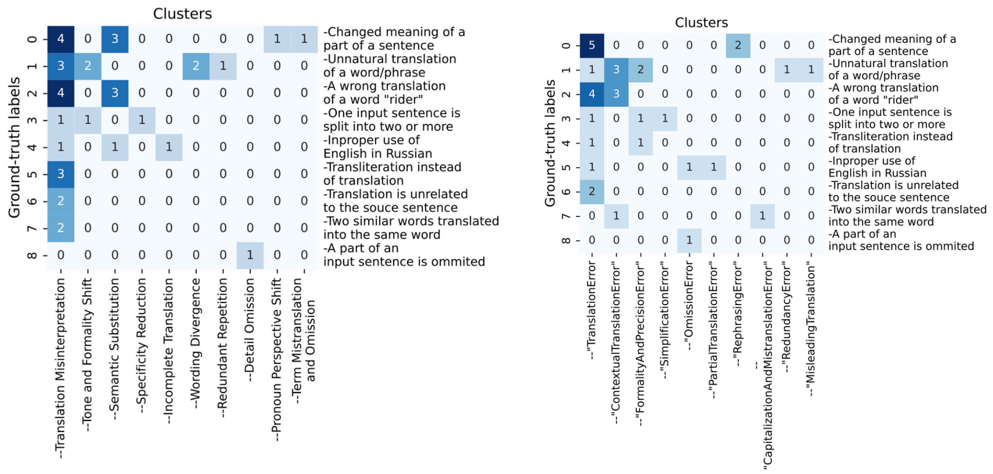
* **Translation Misinterpretation:** The system incorrectly translates specific terms or phrases, leading to inaccurate representations of the original meaning in the generated answer. (20)

* **Semantic Substitution:** The generated translation substitutes critical terms with incorrect alternatives, leading to a change in the intended meaning of the sentence. (7)

* **Tone and Formality Shift:** The generated translation alters the tone and level of formality compared to the ground truth, leading to a failure in matching the expected evaluative criteria. (3)

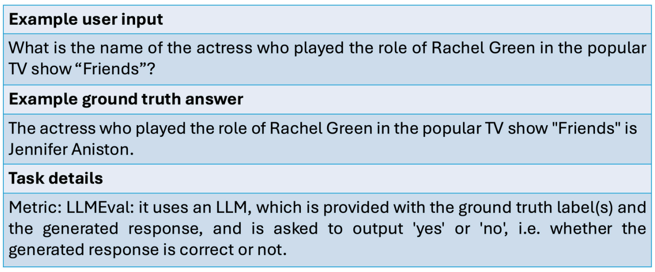
* **Wording Divergence:** The generated answer diverges in wording from the ground truth, causing a mismatch that leads to incorrect evaluation outcomes. (2)

Confusion matrices for issue type clustering for GPT4o (left) and Qwen-2.5-7B (right)



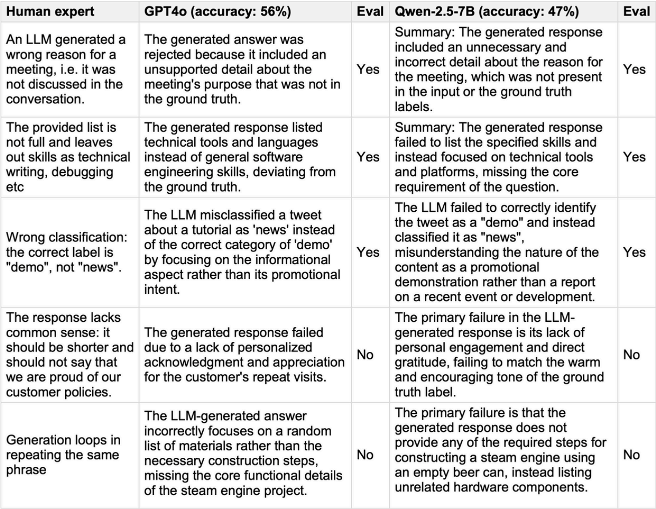
H.5 Instruction following (FLASK dataset)

Task illustration



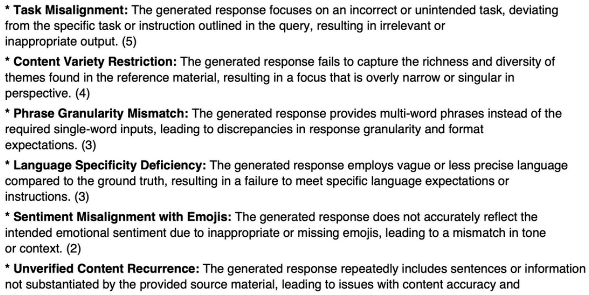
Example user input
What is the name of the actress who played the role of Rachel Green in the popular TV show “Friends”?
Example ground truth answer
The actress who played the role of Rachel Green in the popular TV show "Friends" is Jennifer Aniston.
Task details
Metric: LLMEval: it uses an LLM, which is provided with the ground truth label(s) and the generated response, and is asked to output 'yes' or 'no', i.e. whether the generated response is correct or not.

Examples of per-instance analysis illustration



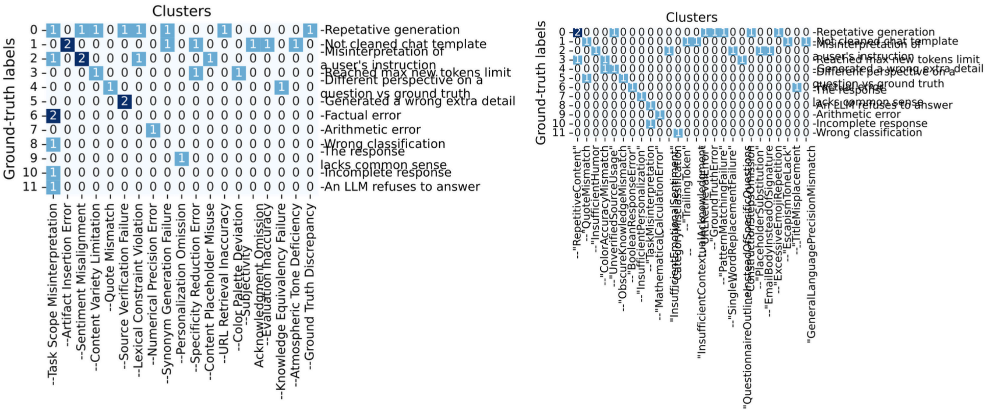
Human expert	GPT4o (accuracy: 56%)	Eval	Qwen-2.5-7B (accuracy: 47%)	Eval
An LLM generated a wrong reason for a meeting, i.e. it was not discussed in the conversation.	The generated answer was rejected because it included an unsupported detail about the meeting's purpose that was not in the ground truth.	Yes	Summary: The generated response included an unnecessary and incorrect detail about the reason for the meeting, which was not present in the input or the ground truth labels.	Yes
The provided list is not full and leaves out skills as technical writing, debugging etc	The generated response listed technical tools and languages instead of general software engineering skills, deviating from the ground truth.	Yes	Summary: The generated response failed to list the specified skills and instead focused on technical tools and platforms, missing the core requirement of the question.	Yes
Wrong classification: the correct label is "demo", not "news".	The LLM misclassified a tweet about a tutorial as 'news' instead of the correct category of 'demo' by focusing on the informational aspect rather than its promotional intent.	Yes	The LLM failed to correctly identify the tweet as a "demo" and instead classified it as "news", misunderstanding the nature of the content as a promotional demonstration rather than a report on a recent event or development.	Yes
The response lacks common sense: it should be shorter and should not say that we are proud of our customer policies.	The generated response failed due to a lack of personalized acknowledgment and appreciation for the customer's repeat visits.	No	The primary failure in the LLM-generated response is its lack of personal engagement and direct gratitude, failing to match the warm and encouraging tone of the ground truth label.	No
Generation loops in repeating the same phrase	The LLM-generated answer incorrectly focuses on a random list of materials rather than the necessary construction steps, missing the core functional details of the steam engine project.	No	The primary failure is that the generated response does not provide any of the required steps for constructing a steam engine using an empty beer can, instead listing unrelated hardware components.	No

An example of a final report generated by LLM-as-a-qualitative-judge (GPT4o)



- **Task Misalignment:** The generated response focuses on an incorrect or unintended task, deviating from the specific task or instruction outlined in the query, resulting in irrelevant or inappropriate output. (5)
- **Content Variety Restriction:** The generated response fails to capture the richness and diversity of themes found in the reference material, resulting in a focus that is overly narrow or singular in perspective. (4)
- **Phrase Granularity Mismatch:** The generated response provides multi-word phrases instead of the required single-word inputs, leading to discrepancies in response granularity and format expectations. (3)
- **Language Specificity Deficiency:** The generated response employs vague or less precise language compared to the ground truth, resulting in a failure to meet specific language expectations or instructions. (3)
- **Sentiment Misalignment with Emojis:** The generated response does not accurately reflect the intended emotional sentiment due to inappropriate or missing emojis, leading to a mismatch in tone or context. (2)
- **Unverified Content Recurrence:** The generated response repeatedly includes sentences or information not substantiated by the provided source material, leading to issues with content accuracy and diversity. (2)
- **Attribute Annotation Error:** The generated response inaccurately assigns or introduces attributes that are absent from the ground truth or input data, leading to errors in attribute extraction or assignment tasks. (2)
- **Evaluation Misjudgment:** The automated evaluation process inaccurately labels a correct generated response as incorrect due to a misalignment between the evaluation criteria and the correct output. (2)
- **Artifact Inclusion:** The generated response contains unintended or residual text artifacts, which disrupt the intended output format and clarity. (2)

Confusion matrices for issue type clustering for GPT4o (left) and Qwen-2.5-7B (right)



H.6 Grade school math (GSM8K dataset)

Task illustration

Example user input

James decides to run 3 sprints 3 times a week. He runs 60 meters each sprint. How many total meters does he run a week?

Example ground truth answer

He sprints 3*3=<<3*3=9>>9 times
So he runs 9*60=<<9*60=540>>540 meters
540

Task details

Metric: Exact Match: it checks if the answer highlighted in \emph{} matches exactly the ground truth answer. We assume that the ground truth labels provided in the dataset are correct and cannot be the source of the pipeline failure.
The LLM answer was obtained using retrieval-augmented generation and chain-of-thought prompt.

Examples of per-instance analysis illustration

Human expert	GPT4o (accuracy: 88%)	Eval	Qwen-2.5-7B (accuracy: 82%)	Eval
A wrong formula in step 3: should be 20 * 2 instead of 20 * 10.	The LLM miscalculated the production cost by incorrectly applying the supply cost to each individual candle, resulting in an erroneous net profit calculation.	Yes	The LLM incorrectly calculated the cost per candle as $10.00 instead of $1.00, leading to an incorrect total cost and net profit.	Yes
The last step misses a multiplication of 126 by 6 and hence outputs a wrong final amount.	The LLM-generated response miscalculated the total annual cost by failing to multiply the discounted monthly rate by the number of months in the second half of the year.	Yes	The LLM failed to correctly compute the total cost for the second half of the year, resulting in an incorrect final total amount of $966 instead of the correct $1596.	Yes
Arithmetic error in step 3: should be 1200 instead of 1500	The LLM miscalculated the enhanced throwing distance with the gemstone, leading to an incorrect safe distance being reported.	Yes	The primary failure in the LLM-generated answer is the incorrect calculation of the throwing distance with the gemstone, resulting in a distance of 1500 feet instead of the correct 1200 feet.	Yes
Difference in formatting between the generated answer and the ground truth answer: The final answer is represented with zeros after the decimal point, .00, while the ground truth answer is represented in an integer format, resulting in an absence of a verbatim match.	The generated answer fails the exact match metric because it unnecessarily includes the verbatim question text, causing a mismatch with the expected concise format of the ground truth.	No	The failure case is due to the lack of explicit highlighting of the final total ($57.00) in the same manner as the ground truth label.	No

An example of a final report generated by LLM-as-a-qualitative-judge (GPT4o)

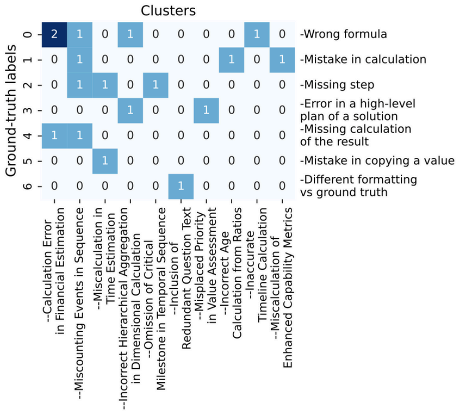
*** Miscounting Events in Sequence:** The LLM makes a mistake in counting or aggregating the number of occurrences in a sequence of events, leading to inaccurate calculations or assertions. (4)

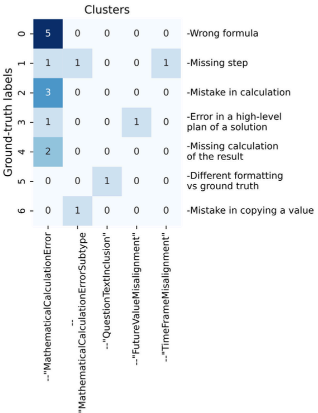
*** Calculation Error in Financial Estimation:** The LLM incorrectly performs computations related to financial metrics, such as costs or profits, due to improper application of inputs or misunderstanding of financial principles. (3)

*** Miscalculation in Time Estimation:** The LLM incorrectly calculates or estimates time-related metrics, such as progress percentages or total time, due to erroneous input handling or flawed understanding of the temporal aspects involved. (2)

*** Incorrect Hierarchical Aggregation in Dimensional Calculation:** The LLM miscalculates dimensional metrics by aggregating levels in the wrong direction, leading to errors in determining total and average values. (2)

Confusion matrices for issue type clustering for GPT4o (left) and Qwen-2.5-7B (right)





H.7 Factoid QA in Russian (MKQA dataset)

Task illustration

Example user input
какой игрок сделал хет-трик в обеих подачах контрольного матча
Example ground truth answer
['Питер Сиддл', 'Джимми Мэтьюз']
Task details
Substring match: it measures whether any of the ground truth responses is included in the model response. We assume that the ground truth labels provided in the dataset are correct and cannot be the source of the pipeline failure. The LLM answer was obtained using retrieval-augmented generation.

Examples of per-instance analysis illustration

Human expert	GPT4o (accuracy: 69%)	Eval	Qwen-2.5-7B (accuracy: 66%)	Eval
The correct answer is conjugated in a sentence resulting in an absence of a verbatim match	The LLM response failed the Substring match test because it included additional context, making it not a strict substring of any of the ground truth labels.	Yes	The failure case is due to the substring match metric not recognizing the ground truth labels "Бразилия" and "Португалия" within the generated answer, even though the answer is factually correct.	Yes
different phrasing of the target date in the generated response compared to the ground-truth label	The generated answer was not accepted due to failing a strict substring match metric, as it included extra textual information alongside the correct date.	Yes	The generated response includes unnecessary details about the publication in different countries, missing the specific date '1997-06-26' that the substring match metric requires.	Yes
different phrasing of the target number (written versus digit) in the generated response compared to the ground-truth label	The LLM response uses words instead of the expected numeric format, causing a mismatch with the ground truth label.	Yes	The generated answer lacks the explicit mention of the number "3.0" which is required by the ground truth label, even though the correct years of victory were provided.	Yes
the target named enity written in English rather than in Russian, while the ground truth is written in Russian	The generated response includes extra information about Instagram's own account, which prevents it from matching the ground truth label focused solely on Cristiano Ronaldo.	No	The generated response included unnecessary details about the Instagram account's follower count, which caused it to fail the substring match criterion despite being factually correct.	No
different transliteration of the target named entity (into the Cyrillic alphabet) in the generated response compared to the ground-truth label	The generated response only contains one of the two required ground truth names, failing the substring match metric.	No	The generated answer includes unnecessary details that are not present in the ground truth labels, leading to a mismatch in the substring match evaluation.	No

An example of a final report generated by LLM-as-a-qualitative-judge (GPT4o)

* **Non-Canonical Output Error:** The generated response includes extraneous words or adopts an unconventional format that deviates from the expected canonical representation, hindering accurate evaluation against the ground truth. (7)

* **Missing Key Information Error:** The generated response omits essential information that is present in the ground truth, resulting in an incomplete answer. (3)

* **Name Variation and Extraneous Context Error:** The generated response contains minor variations in names and includes additional contextual information not present in the ground truth, leading to a mismatch in evaluation. (3)

* **Format Mismatch Error:** The generated response uses an incorrect format, such as substituting words for the expected numeric format, leading to a discrepancy with the ground truth label. (2)

* **Precision Error in Temporal Data:** The generated response provides an approximate temporal detail instead of the exact value required, leading to a mismatch with the specific ground truth information. (2)

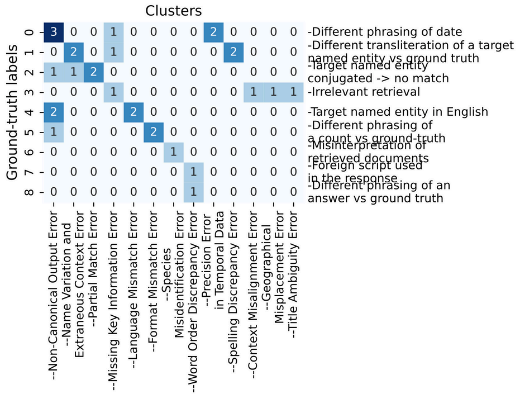
* **Word Order Discrepancy Error:** The generated response contains the correct information but in a different word order than the ground truth, resulting in a non-match by the substring match metric. (2)

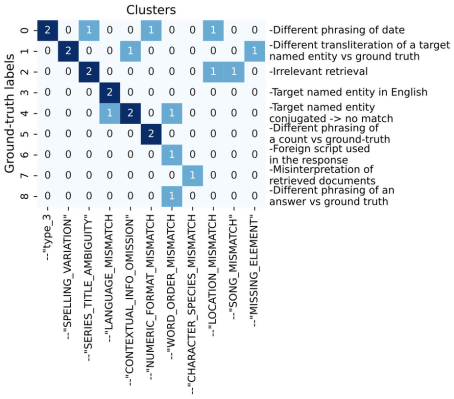
* **Spelling Discrepancy Error:** The generated response contains a spelling variation in critical names or terms that results in a failure to match the ground truth label, despite conveying the intended information. (2)

* **Language Mismatch Error:** The generated response is in a different language than the ground truth, causing a failure in matching the correct answer in automatic evaluation. (2)

* **Partial Match Error:** The generated response includes the correct answer within a larger text, leading to a failure in exact substring matching during evaluation. (2)

Confusion matrices for issue type clustering for GPT4o (left) and Qwen-2.5-7B (right)





H.8 Biomedical QA (BioASQ dataset)

Task illustration

Example user input

Which protein interacts with the Ragulator-RAG GTPases to control mTOR activity?

Example ground truth answer

['Amino Acid-DepeLysosomal Membrane Protein SLC38A9']

Task details

Metric: Substring match: it measures whether any of the ground truth responses is included in the model response. We assume that the ground truth labels provided in the dataset are correct and cannot be the source of the pipeline failure.
The LLM answer was obtained using retrieval-augmented generation.

Examples of per-instance analysis illustration

Human expert	GPT4o (accuracy: 67%)	Eval	Qwen-2.5-7B (accuracy: 63%)	Eval
The generated response as well as retrieved documents contain a different list of deseases than the ground truth list	The generated answer includes 'multiple sclerosis', a disease not present in the ground truth labels, causing a mismatch.	Yes	The primary failure in this example is the omission of "obsessive-compulsive disorder" from the list of diseases that can be treated with focused ultrasound thalamotomy.	Yes
match metric error: correct answer is judged wrong because of the absence of the verbatim match	The generated answer failed the substring match metric because it listed the correct tools separately with additional details, rather than as a continuous substring matching the ground truth label.	Yes	The failure case arises from the detailed description of 'EPD' in the generated answer, which does not strictly match the concise format required by the ground truth label.	Yes
Misinterpretation of the retrieved document: Document 4 does not mention atemoya juice while the generated response points to this document as a reference	The LLM-generated answer contradicts the ground truth 'no' due to reliance on potentially misleading retrieved information.	Yes	The LLM-generated answer incorrectly affirms that atemoya juice inhibits the CYP3A4 enzyme, contradicting the ground truth label which states 'no'.	Yes
Ambiguous question: the generated response provides one way to answer the question and the ground truth label provides another way.	The LLM-generated response failed to include the exact term 'Leishmania species' required for a substring match with the ground truth.	No	The generated answer failed to include the term "Leishmania Species," which is required by the ground truth label for the substring match to succeed.	No
The generated response is cropped	The LLM-generated answer failed due to being too detailed and not exactly matching the concise ground truth label 'yes' required by the evaluation metric.	No	The generated response fails to directly affirm the existence of data supporting the improvement of symptom severity in ALS patients through TRH administration, as required by the ground truth label.	No

An example of a final report generated by LLM-as-a-qualitative-judge (GPT4o)

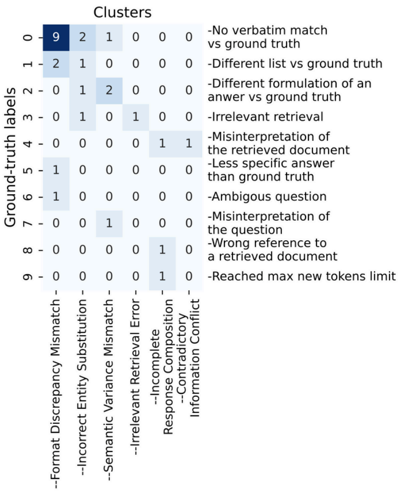
* **Format Discrepancy Mismatch:** The generated answer fails to match the ground truth due to differences in formatting or wording while retaining the same semantic meaning. (13)

* **Incorrect Entity Substitution:** The generated answer inaccurately replaces or incorporates an incorrect entity or detail, deviating from the ground truth despite correct interpretation of the retrieved content. (5)

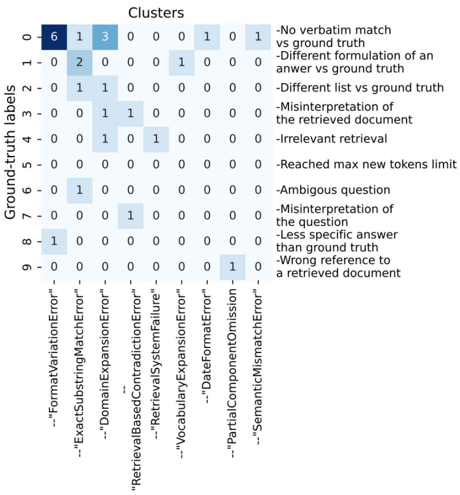
* **Semantic Variance Mismatch:** The generated answer diverges in semantic meaning from the ground truth, leading to a mismatch that goes beyond mere formatting or wording differences. (4)

* **Incomplete Response Composition:** The generated answer omits required components of the ground truth, resulting in a response that only partially addresses the complete expected answer. (3)

Confusion matrices for issue type clustering for GPT4o (left) and Qwen-2.5-7B (right)



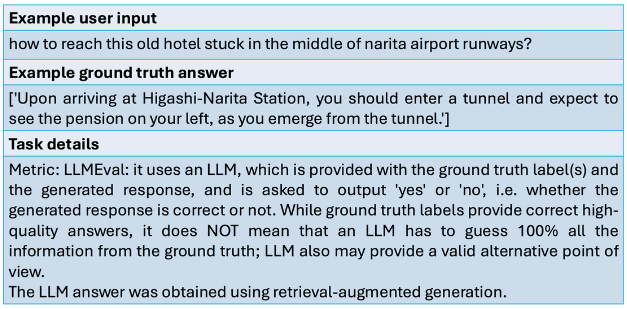
Ground-truth labels	--Format Discrepancy Mismatch	--Incorrect Entity Substitution	--Semantic Variance Mismatch	--Irrelevant Retrieval Error	--Incomplete Response Composition	--Contradictory Information Conflict	
0	9	2	1	0	0	0	-No verbatim match vs ground truth
1	2	1	0	0	0	0	-Different list vs ground truth
2	0	1	2	0	0	0	-Different formulation of an anwer vs ground truth
3	0	1	0	1	0	0	-Irrelevant retrieval
4	0	0	0	0	1	1	-Misinterpretation of the retrieved document
5	1	0	0	0	0	0	-Less specific answer than ground truth
6	1	0	0	0	0	0	-Ambigous question
7	0	0	1	0	0	0	-Misinterpretation of the question
8	0	0	0	0	1	0	-Wrong reference to a retrieved document
9	0	0	0	0	1	0	-Reached max new tokens limit



Ground-truth labels	--"FormatVariationError"	--"ExactSubstringMatchError"	--"DomainExpansionError"	--"RetrievalBasedContradictionError"	--"RetrievalSystemFailure"	--"VocabularyExpansionError"	--"DateFormatError"	--"PartialComponentOmission"	--"SemanticMismatchError"	
0	6	1	3	0	0	0	1	0	1	-No verbatim match vs ground truth
1	0	2	0	0	0	1	0	0	0	-Different formulation of an anwer vs ground truth
2	0	1	1	0	0	0	0	0	0	-Different list vs ground truth
3	0	0	1	1	0	0	0	0	0	-Misinterpretation of the retrieved document
4	0	0	1	0	1	0	0	0	0	-Irrelevant retrieval
5	0	0	0	0	0	0	0	0	0	-Reached max new tokens limit
6	0	1	0	0	0	0	0	0	0	-Ambigous question
7	0	0	0	1	0	0	0	0	0	-Misinterpretation of the question
8	1	0	0	0	0	0	0	0	0	-Less specific answer than ground truth
9	0	0	0	0	0	0	0	1	0	-Wrong reference to a retrieved document

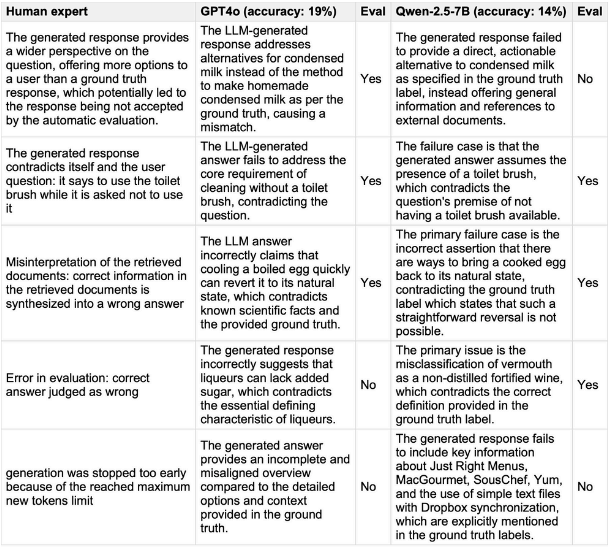
H.9 Lifestyle forum QA (RobustQA dataset)

Task illustration



Example user input
how to reach this old hotel stuck in the middle of narita airport runways?
Example ground truth answer
['Upon arriving at Higashi-Narita Station, you should enter a tunnel and expect to see the pension on your left, as you emerge from the tunnel.']
Task details
Metric: LLMEval: it uses an LLM, which is provided with the ground truth label(s) and the generated response, and is asked to output 'yes' or 'no', i.e. whether the generated response is correct or not. While ground truth labels provide correct high-quality answers, it does NOT mean that an LLM has to guess 100% all the information from the ground truth; LLM also may provide a valid alternative point of view. The LLM answer was obtained using retrieval-augmented generation.

Examples of per-instance analysis illustration



Human expert	GPT4o (accuracy: 19%)	Eval	Qwen-2.5-7B (accuracy: 14%)	Eval
The generated response provides a wider perspective on the question, offering more options to a user than a ground truth response, which potentially led to the response being not accepted by the automatic evaluation.	The LLM-generated response addresses alternatives for condensed milk instead of the method to make homemade condensed milk as per the ground truth, causing a mismatch.	Yes	The generated response failed to provide a direct, actionable alternative to condensed milk as specified in the ground truth label, instead offering general information and references to external documents.	No
The generated response contradicts itself and the user question: it says to use the toilet brush while it is asked not to use it	The LLM-generated answer fails to address the core requirement of cleaning without a toilet brush, contradicting the question.	Yes	The failure case is that the generated answer assumes the presence of a toilet brush, which contradicts the question's premise of not having a toilet brush available.	Yes
Misinterpretation of the retrieved documents: correct information in the retrieved documents is synthesized into a wrong answer	The LLM answer incorrectly claims that cooling a boiled egg quickly can revert it to its natural state, which contradicts known scientific facts and the provided ground truth.	Yes	The primary failure case is the incorrect assertion that there are ways to bring a cooked egg back to its natural state, contradicting the ground truth label which states that such a straightforward reversal is not possible.	Yes
Error in evaluation: correct answer judged as wrong	The generated response incorrectly suggests that liqueurs can lack added sugar, which contradicts the essential defining characteristic of liqueurs.	No	The primary issue is the misclassification of vermouth as a non-distilled fortified wine, which contradicts the correct definition provided in the ground truth label.	Yes
generation was stopped too early because of the reached maximum new tokens limit	The generated answer provides an incomplete and misaligned overview compared to the detailed options and context provided in the ground truth.	No	The generated response fails to include key information about Just Right Menus, MacGourmet, SousChef, Yum, and the use of simple text files with Dropbox synchronization, which are explicitly mentioned in the ground truth labels.	No

An example of a final report generated by LLM-as-a-qualitative-judge (GPT4o)

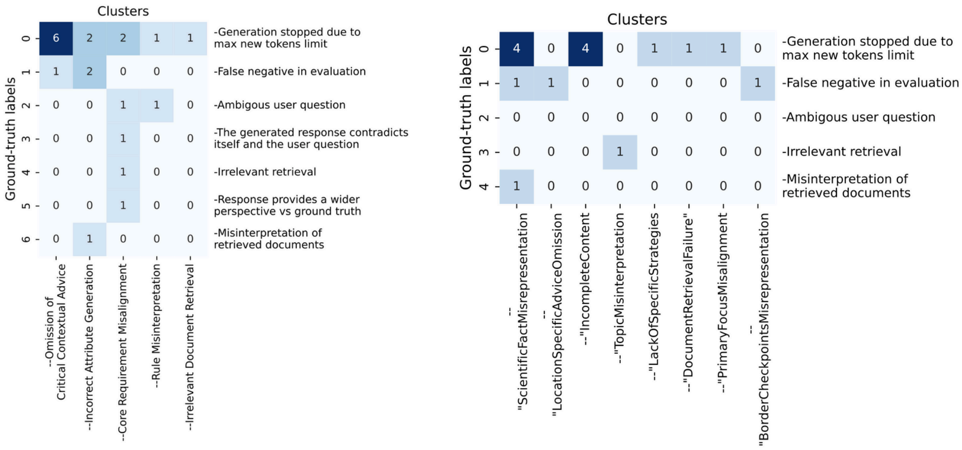
* **Omission of Critical Contextual Advice:** The response fails to include essential contextual advice or recommendations relevant to the specific needs or circumstances of the subject matter, focusing instead on generic or peripheral information. (7)

* **Core Requirement Misalignment:** The response fails to align with the central requirement or intent of the query, resulting in a response that contradicts or overlooks the primary objective sought by the user. (6)

* **Incorrect Attribute Generation:** The generated response provides incorrect or contradictory information about a defining attribute of the discussed subject, conflicting with established facts. (5)

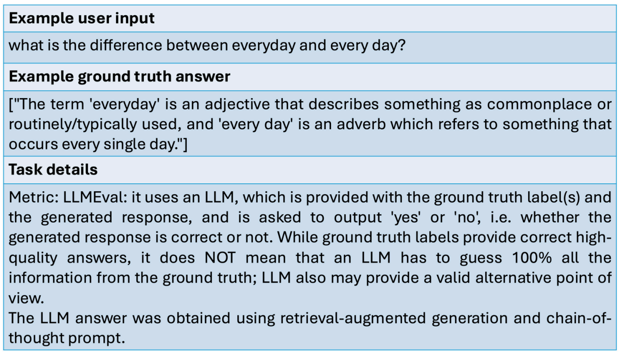
* **Rule Misinterpretation:** The response is based on a misinterpretation of formal rules or guidelines, leading to an inaccurate conclusion or recommendation. (2)

Confusion matrices for issue type clustering for GPT4o (left) and Qwen-2.5-7B (right)



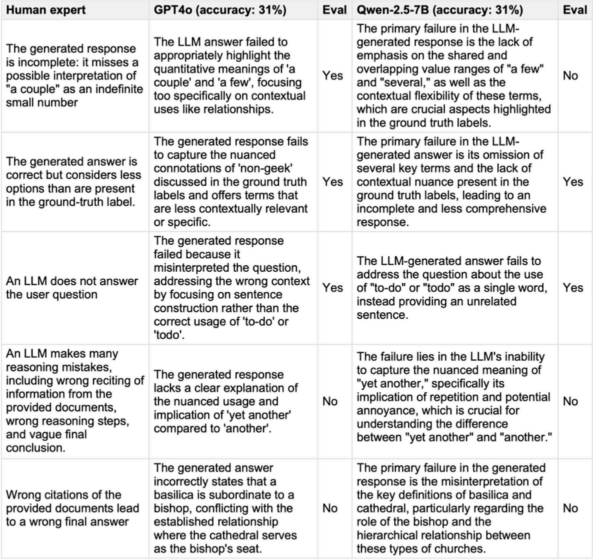
H.10 Writing forum QA (RobustQA dataset)

Task illustration



Example user input
what is the difference between everyday and every day?
Example ground truth answer
["The term 'everyday' is an adjective that describes something as commonplace or routinely/typically used, and 'every day' is an adverb which refers to something that occurs every single day."]
Task details
Metric: LLMEval: it uses an LLM, which is provided with the ground truth label(s) and the generated response, and is asked to output 'yes' or 'no', i.e. whether the generated response is correct or not. While ground truth labels provide correct high-quality answers, it does NOT mean that an LLM has to guess 100% all the information from the ground truth; LLM also may provide a valid alternative point of view. The LLM answer was obtained using retrieval-augmented generation and chain-of-thought prompt.

Examples of per-instance analysis illustration



Human expert	GPT4o (accuracy: 31%)	Eval	Qwen-2.5-7B (accuracy: 31%)	Eval
The generated response is incomplete: it misses a possible interpretation of "a couple" as an indefinite small number	The LLM answer failed to appropriately highlight the quantitative meanings of 'a couple' and 'a few', focusing too specifically on contextual uses like relationships.	Yes	The primary failure in the LLM-generated response is the lack of emphasis on the shared and overlapping value ranges of "a few" and "several," as well as the contextual flexibility of these terms, which are crucial aspects highlighted in the ground truth labels.	No
The generated answer is correct but considers less options than are present in the ground-truth label.	The generated response fails to capture the nuanced connotations of 'non-geek' discussed in the ground truth labels and offers terms that are less contextually relevant or specific.	Yes	The primary failure in the LLM-generated answer is its omission of several key terms and the lack of contextual nuance present in the ground truth labels, leading to an incomplete and less comprehensive response.	Yes
An LLM does not answer the user question	The generated response failed because it misinterpreted the question, addressing the wrong context by focusing on sentence construction rather than the correct usage of 'to-do' or 'todo'.	Yes	The LLM-generated answer fails to address the question about the use of "to-do" or "todo" as a single word, instead providing an unrelated sentence.	Yes
An LLM makes many reasoning mistakes, including wrong reciting of information from the provided documents, wrong reasoning steps, and vague final conclusion.	The generated response lacks a clear explanation of the nuanced usage and implication of 'yet another' compared to 'another'.	No	The failure lies in the LLM's inability to capture the nuanced meaning of "yet another," specifically its implication of repetition and potential annoyance, which is crucial for understanding the difference between "yet another" and "another."	No
Wrong citations of the provided documents lead to a wrong final answer	The generated answer incorrectly states that a basilica is subordinate to a bishop, conflicting with the established relationship where the cathedral serves as the bishop's seat.	No	The primary failure in the generated response is the misinterpretation of the key definitions of basilica and cathedral, particularly regarding the role of the bishop and the hierarchical relationship between these types of churches.	No

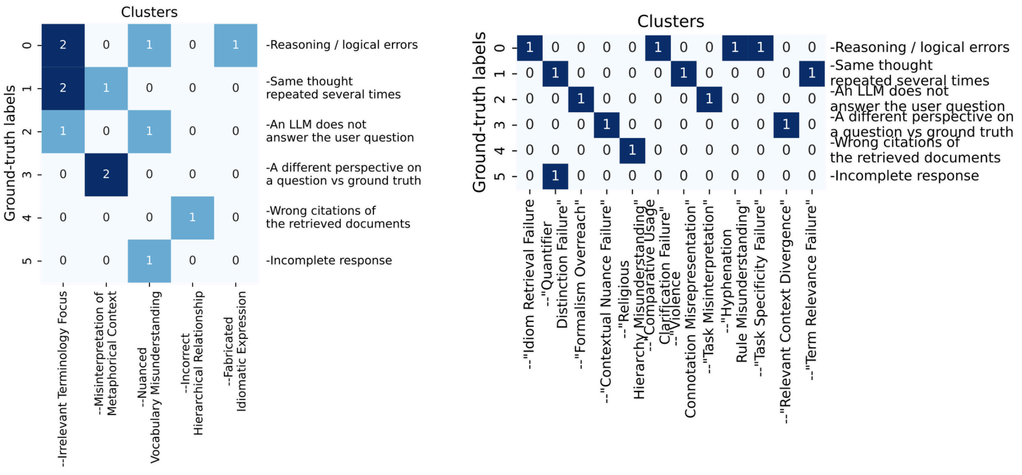
An example of a final report generated by LLM-as-a-qualitative-judge (GPT4o)

* **Irrelevant Terminology Focus:** The generated response incorrectly prioritizes irrelevant linguistic details over key terms or concepts, resulting in an inaccurate representation of the main subject or context. (5)

* **Misinterpretation of Metaphorical Context:** The generated response misinterprets metaphorical or emotional nuances in retrieved documents, resulting in a divergence from the intended thematic context. (3)

* **Nuanced Vocabulary Misunderstanding:** The generated response fails to articulate subtle distinctions in vocabulary usage, leading to a shallow or misleading interpretation of context-specific language nuances. (3)

Confusion matrices for issue type clustering for GPT4o (left) and Qwen-2.5-7B (right)



H.11 Search engine queries (SearchQA dataset)

Task illustration

Example user input
I'm going to hop in my Volvo for a trip from Jokkmokk to Skovde in this country
Example ground truth answer
['Sweden']
Task details
Metric: LLMEval: it uses an LLM, which is provided with the ground truth label(s) and the generated response, and is asked to output 'yes' or 'no', i.e. whether the generated response is correct or not. We assume that the ground truth labels provided in the dataset are correct and cannot be the source of the pipeline failure. The LLM answer was obtained using retrieval-augmented generation.

Examples of per-instance analysis illustration

Human expert	GPT4o (accuracy: 54%)	Eval	Qwen-2.5-7B (accuracy: 8%)	Eval
the retrieved documents do not contain the correct answer	The retrieved documents did not contain the correct information about Australia's 1999 referendum on the British Commonwealth, resulting in an incorrect generated answer.	Yes	The generated answer failed to correctly identify the specific collective entity (the British Commonwealth) from which Australia was considering withdrawal, instead providing a vague description of a collective security system.	No
an LLM repeats a question without answering it	The generated response fails to answer the question as it repeats the question verbatim without providing the required answer 'Sweden'.	Yes	The LLM failed to extract and incorporate the relevant information from the context into its response, specifically missing the identification of the country as 'Sweden'.	No
a mistake in a retrieved document which is propagated to the llm response	The retrieval step supplied incorrect information about Felipe Calderon, erroneously linking him to Nicaragua, causing the LLM to generate an incorrect answer.	Yes	The primary issue is the incorrect identification of the country associated with President Felipe Calderón, likely due to the LLM retrieving and misinterpreting information that linked him to Nicaragua instead of Mexico.	Yes
Unclear user question	The LLM failed by not providing the author's name and misinterpreting the context, leading to incorrect pronoun usage.	No	The LLM failed to correctly attribute the drowning incident to the right character, misquoting "Roberta Alden" instead of "Grace Brown."	No
error of LLM evaluation, because the provided answer is correct	The generated answer unnecessarily introduces doubt about the correct information, causing it to be considered incorrect.	No	The generated response included unnecessary details and did not directly state the name "William Blake", leading to it being marked incorrect by the evaluation metric.	No

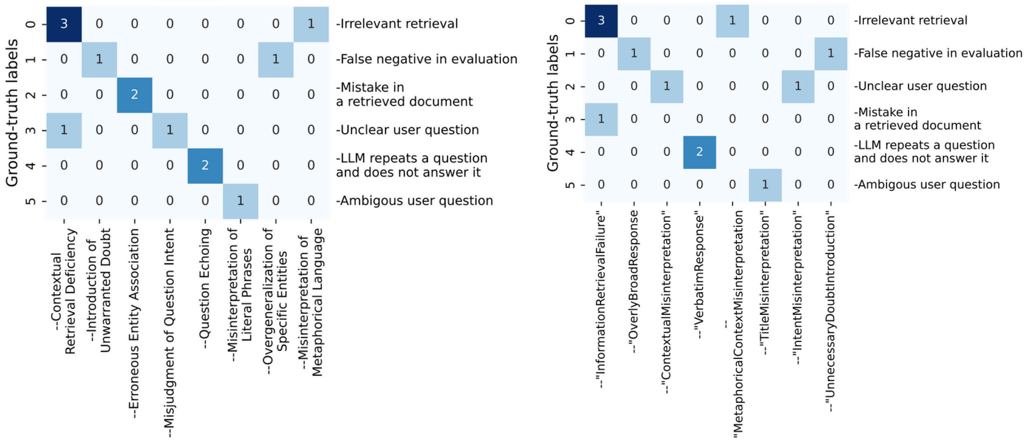
An example of a final report generated by LLM-as-a-qualitative-judge (GPT4o)

* **Contextual Retrieval Deficiency:** The generated response fails due to a lack of sufficient context from the retrieved documents, resulting in a misunderstanding of the question's requirements. (4)

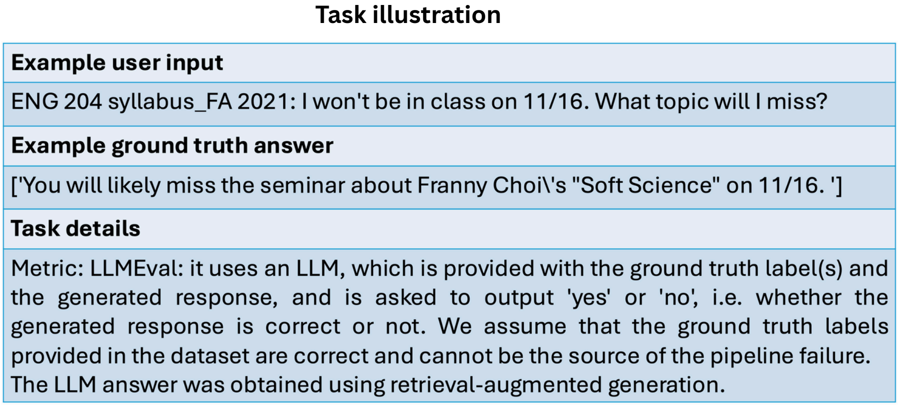
* **Erroneous Entity Association:** The retrieval step provides incorrect associations between entities, leading to the generation of inaccurate responses due to misinformation. (2)

* **Question Echoing:** The generated response echoes or repeats the question without providing a substantive answer, resulting in a failure to address the query with the required information. (2)

Confusion matrices for issue type clustering for GPT4o (left) and Qwen-2.5-7B (right)



H.12 Educational QA (SyllabusQA dataset)



Task illustration

Example user input

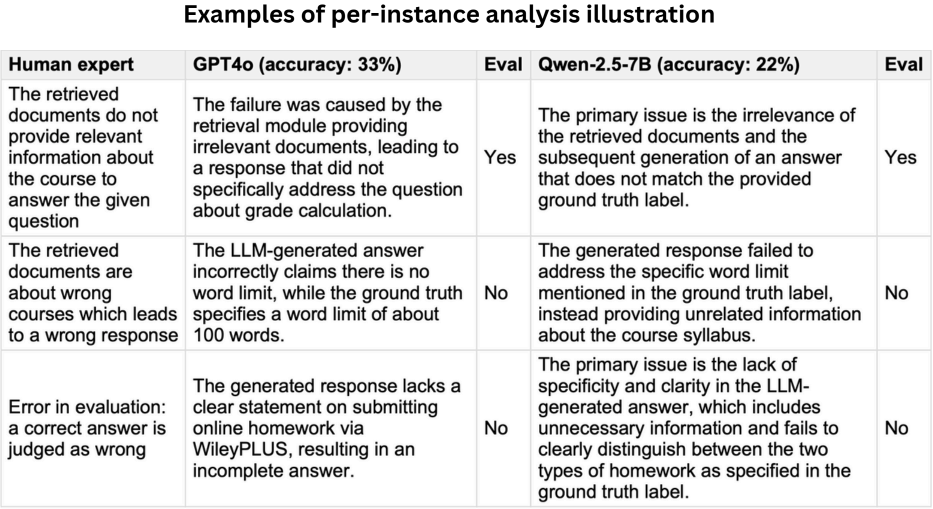
ENG 204 syllabus_FA 2021: I won't be in class on 11/16. What topic will I miss?

Example ground truth answer

['You will likely miss the seminar about Franny Choi\'s "Soft Science" on 11/16. ']

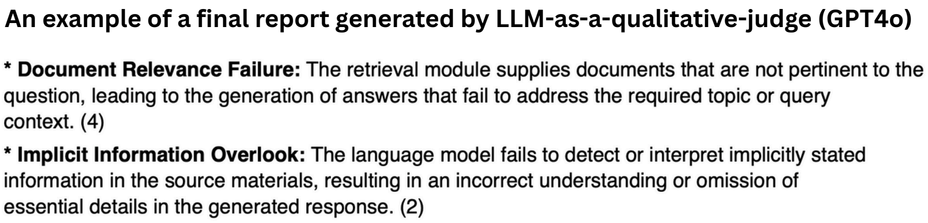
Task details

Metric: LLMEval: it uses an LLM, which is provided with the ground truth label(s) and the generated response, and is asked to output 'yes' or 'no', i.e. whether the generated response is correct or not. We assume that the ground truth labels provided in the dataset are correct and cannot be the source of the pipeline failure. The LLM answer was obtained using retrieval-augmented generation.



Examples of per-instance analysis illustration

Human expert	GPT4o (accuracy: 33%)	Eval	Qwen-2.5-7B (accuracy: 22%)	Eval
The retrieved documents do not provide relevant information about the course to answer the given question	The failure was caused by the retrieval module providing irrelevant documents, leading to a response that did not specifically address the question about grade calculation.	Yes	The primary issue is the irrelevance of the retrieved documents and the subsequent generation of an answer that does not match the provided ground truth label.	Yes
The retrieved documents are about wrong courses which leads to a wrong response	The LLM-generated answer incorrectly claims there is no word limit, while the ground truth specifies a word limit of about 100 words.	No	The generated response failed to address the specific word limit mentioned in the ground truth label, instead providing unrelated information about the course syllabus.	No
Error in evaluation: a correct answer is judged as wrong	The generated response lacks a clear statement on submitting online homework via WileyPLUS, resulting in an incomplete answer.	No	The primary issue is the lack of specificity and clarity in the LLM-generated answer, which includes unnecessary information and fails to clearly distinguish between the two types of homework as specified in the ground truth label.	No

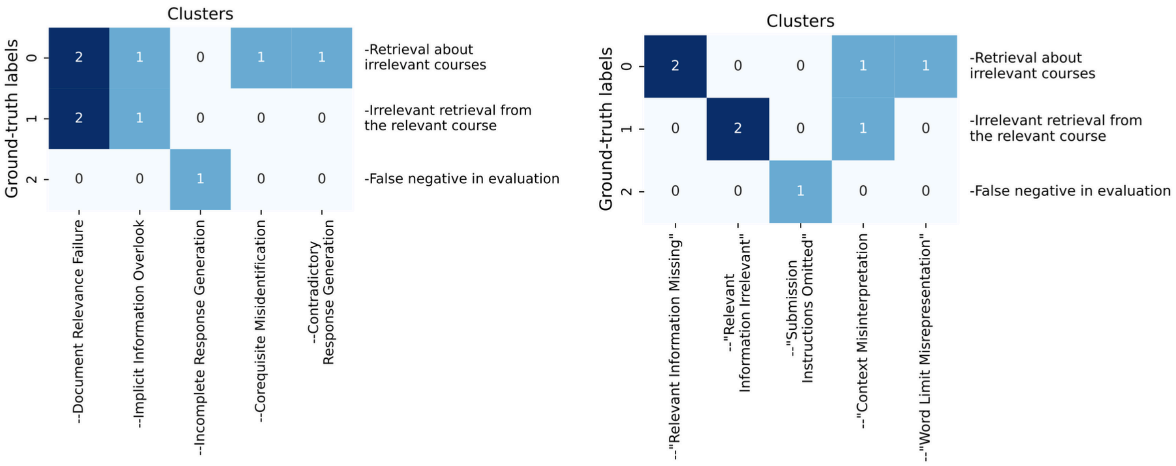


An example of a final report generated by LLM-as-a-qualitative-judge (GPT4o)

*** Document Relevance Failure:** The retrieval module supplies documents that are not pertinent to the question, leading to the generation of answers that fail to address the required topic or query context. (4)

*** Implicit Information Overlook:** The language model fails to detect or interpret implicitly stated information in the source materials, resulting in an incorrect understanding or omission of essential details in the generated response. (2)

Confusion matrices for issue type clustering for GPT4o (left) and Qwen-2.5-7B (right)


Clusters
Ground-truth labels
-Retrieval about irrelevant courses
-Irrelevant retrieval from the relevant course
-False negative in evaluation
--Document Relevance Failure
--Implicit Information Overlook
--Incomplete Response Generation
--Corequisite Misidentification
--Contradictory Response Generation
Clusters
Ground-truth labels
-Retrieval about irrelevant courses
-Irrelevant retrieval from the relevant course
-False negative in evaluation
--"Relevant Information Missing"
--"Relevant Information Irrelevant"
--"Submission Instructions Omitted"
--"Context Misinterpretation"
--"Word Limit Misrepresentation"