information

MDPI

Article

KGEval: Evaluating Scientific Knowledge Graphs with Large Language Models

Vladyslav Nechakhin [1,*], Jennifer D'Souza [1], Steffen Eger [2] and Sören Auer [1]

[1] Leibniz Information Centre for Science and Technology, 30167 Hannover, Germany
[2] Natural Language Learning & Generation (NLLG), University of Technology Nuremberg (UTN), 90461 Nuremberg, Germany
* Correspondence: vlad.nechakhin@tib.eu

Abstract

This paper explores the novel application of large language models (LLMs) as evaluators for structured scientific summaries—a task where traditional natural language evaluation metrics may not readily apply. Leveraging the Open Research Knowledge Graph (ORKG) as a repository of human-curated properties, we augment a gold-standard dataset by generating corresponding properties using three distinct LLMs—Llama, Mistral, and Qwen—under three contextual settings: context-lean (research problem only), context-rich (research problem with title and abstract), and context-dense (research problem with multiple similar papers). To assess the quality of these properties, we employ LLM evaluators (Deepseek, Mistral, and Qwen) to rate them on criteria, including similarity, relevance, factuality, informativeness, coherence, and specificity. This study addresses key research questions: How do LLM-as-a-judge rubrics transfer to the evaluation of structured summaries? How do LLM-generated properties compare to human-annotated ones? What are the performance differences among various LLMs? How does the amount of contextual input affect the generation quality? The resulting evaluation framework, KGEval, offers a customizable approach that can be extended to diverse knowledge graphs and application domains. Our experimental findings reveal distinct patterns in evaluator biases, contextual sensitivity, and inter-model performance, thereby highlighting both the promise and the challenges of integrating LLMs into structured science evaluation.

Keywords: LLMs; structured science; evaluation framework; KGEval

1. Introduction

Knowledge bases such as the Open Research Knowledge Graph (ORKG) [1] are crucial for making scientific findings FAIR (findable, accessible, interoperable, and reusable) [2]. By providing structured summaries of research contributions, these KGs enable efficient comparison and retrieval of scholarly work. However, populating such knowledge graphs is inherently costly and time-consuming, as it relies heavily on manual curation by domain experts.

Large language models (LLMs) have shown great promise in automating the construction of structured representations, potentially alleviating the burden of manual annotation [3]. Despite their potential, evaluating the quality of LLM-generated outputs in the scientific domain poses unique challenges. Traditional NLP metrics such as BLEU [4] and ROUGE [5] are primarily designed to assess surface-level text matching and do not capture

Academic Editor: Katsuhide Fujita

Received: 25 November 2025
Revised: 27 December 2025
Accepted: 31 December 2025
Published: 3 January 2026

deeper semantic meaning or the task-specific nuances required for structured science summarization. For example, while these metrics can measure word overlap between generated and reference texts, they fail to assess whether the generated properties accurately and comprehensively reflect the underlying research problem.

To address these challenges, we introduce **KGEval**, a novel framework that leverages LLMs both as generators and as evaluators (i.e., as LLM-as-a-judge [6]) of structured scientific summaries. KGEval is built with a modular architecture that enables the interchangeability of LLMs and allows for customization of prompts and evaluation criteria. This flexibility is essential for systematically investigating how various context types and evaluation strategies influence the quality of generated properties in a domain as complex as scientific research.

Specifically, our work investigates the following research questions:

RQ1. Transferability of Qualitative Rubrics: How effectively do LLM-as-a-judge evaluation rubrics capture the quality of structured summaries, given that these outputs lack the conventional sentence structures found in natural language?

RQ2. Comparison with Human Annotations: How do the properties generated by LLMs compare to human-annotated properties in terms of relevance, consistency, and other evaluation criteria?

RQ3. Inter-Model Performance: How do different LLMs (e.g., Qwen, Mistral, and Llama) perform in generating properties, and how does their performance compare?

RQ4. Impact of Context: How does the amount and type of contextual input (research problem only vs. research problem with one abstract vs. research problem with multiple abstracts) affect the quality of the generated properties?

In this paper, we focus primarily on the evaluation of structured outputs. By leveraging LLMs as both generators and evaluators, KGEval systematically assesses the quality of generated scientific properties while addressing challenges such as scientific domain complexity and the high cost of manual evaluation. Our contributions are threefold. First, we introduce a robust evaluation framework that repurposes LLMs as evaluators, overcoming the limitations of traditional metrics. Second, we provide a comparative analysis between LLM-generated properties and expert-annotated ORKG properties, highlighting both strengths and areas for improvement. Third, we examine how varying contextual inputs influence the performance of different LLMs, thereby offering insights into the scalability and adaptability of automated structured science summarization.

In the following sections, we detail the KGEval framework, describe our dataset and experimental setup, and present an in-depth analysis of our results.

2. Related Work

In the NLP community, developing evaluation metrics that reliably measure the quality of tasks such as translation or summarization has long been a fundamental concern. In recent decades, a plethora of different paradigms have been suggested. These range from (1) lexical overlap metrics such as BLEU [4] and ROUGE [5], which are inherently limited, to (2) semantic similarity metrics like BERTScore [7] and MoverScore [8]; (3) text generation-based metrics such as BARTScore [9] and PRISM [10]; (4) natural language inference metrics like MENLI [11], which promise to increase robustness; and (5) prompting-based metrics such as GEMBA [12] which rely on LLMs and their prompts for judging the quality of outputs. While these metrics have been explored for 'standard' tasks like summarization and machine translation, their potential for evaluating structured science summaries remains fundamentally underexplored. In this work, we fill this gap, focusing primarily on prompt-based metrics to evaluate structured science summaries.

Several recent studies have adopted the LLM-as-a-judge paradigm to assess generated content by correlating LLM evaluations with human judgments using pairwise preference evaluations [13–17]. Building on these methods, some approaches have incorporated rubric-based techniques—such as G-Eval [17] for summarization and GPTScore [18] for flexible prompt-based evaluation—to capture nuanced aspects of generated text. In addition, frameworks such as FLASK [19] and Prometheus [20] have advanced the state of the art by emphasizing fine-grained rubrics that assess robustness, correctness, efficiency, factuality, and readability. Together, these studies underscore the evolving landscape of LLM-based evaluation frameworks and motivate our extension of the paradigm to the domain of structured science summarization.

Our previous work [21] laid the groundwork by exploring the feasibility of using LLMs to recommend research properties for structured science summarization in the ORKG. That study employed methods such as semantic alignment and deviation assessments, fine-grained property-to-dimension mappings, embeddings-based evaluations, and human surveys to compare LLM-generated dimensions with expert-curated ORKG properties. Building upon these findings, our current work extends the LLM-as-a-judge paradigm through the KGEval framework, integrating both generation and evaluation in a unified system while systematically examining evaluator biases and context effects.

In summary, while a range of evaluation metrics and frameworks have been proposed in the literature, our work contributes by extending prompting-based evaluation to the domain of structured science summarization. By building on the advancements in LLM-based evaluation rubrics and our previous findings, we provide a robust, open-science framework that is adaptable to diverse scientific KGs.

3. The KGEval Framework

In this section, we present KGEval, a modular framework designed to both generate and evaluate structured scientific properties. The framework is built around two primary modules: an LLM generator and an LLM evaluator. This modular design enables KGEval to handle various input contexts and property sources flexibly, making it adaptable to a wide range of KGs and evaluation tasks.

3.1. Framework Overview and Workflow

KGEval is structured as a two-module system that operates in a sequential yet modular fashion. The first module, the LLM generator, accepts diverse forms of input context, such as research questions, abstracts, full papers, articles, or even multiple related papers. Regardless of the context—whether *context-lean* (research problem only), *-rich* (research problem with title and abstract), or *-dense* (research problem with multiple abstracts)—the generator utilizes customizable prompts (which remain consistent across scenarios, with only the input context varying) to produce structured representations, hereafter referred to as *properties*. Once these properties are generated, they are forwarded to the second module, the LLM evaluator.

The evaluator module is designed to assess the quality of the properties based on a comprehensive, unified prompt that incorporates multiple qualitative criteria. This module accepts properties generated by the LLM generator as well as those obtained from external sources, such as human-annotated entries from a KG (e.g., ORKG). The evaluator then outputs a quantitative score reflecting the quality of the input properties according to criteria such as similarity, relevance, factuality, informativeness, coherence, and specificity. Both modules leverage a shared LLM management system that supports various models (e.g., Deepseek, Llama, Mistral, and Qwen), which can be run locally or accessed via API. Figure 1 illustrates the KGEval pipeline, showing how context (research question, abstract,

and papers) and prompts feed into the generator, how the generated properties and human-annotated properties (from a KG) are then evaluated against defined criteria, and how the evaluator produces a final score.

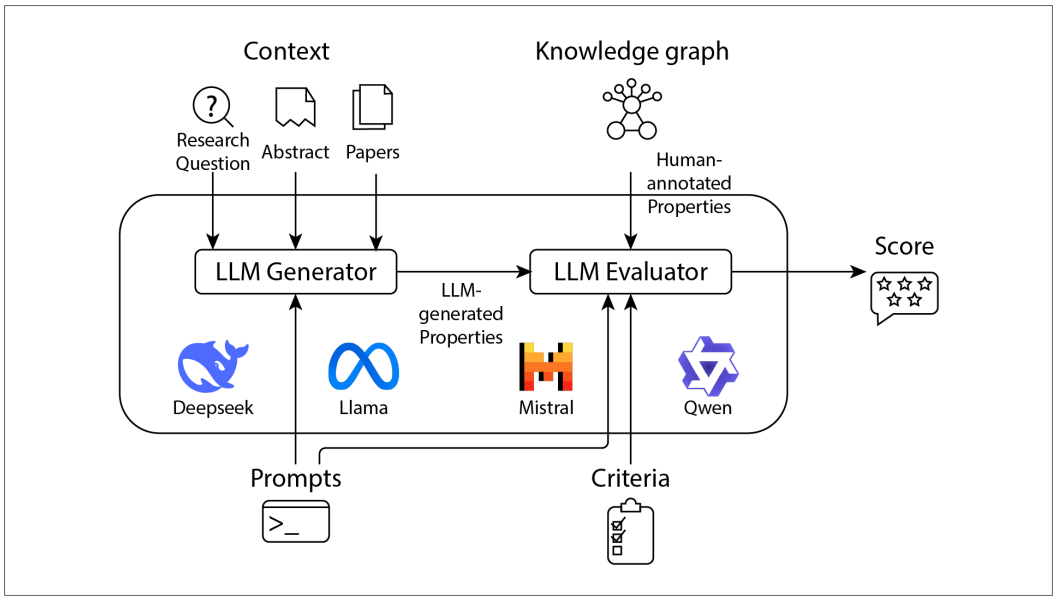

Figure 1. Overview of the KGEval workflow.

3.2. Evaluation Scenarios and Criteria

KGEval supports two primary evaluation scenarios: *direct assessment* and *pairwise ranking*. In the direct assessment scenario, a single set of properties—either generated by the LLM or curated by humans—is evaluated against the qualitative criteria. This process yields individual scores that reflect the properties' relevance, factuality, informativeness, coherence, and specificity. In contrast, the pairwise ranking scenario directly compares two sets of properties. For example, KGEval compares properties generated using a research problem alone (context-lean) against those generated with additional contextual information (rich or dense context), as well as comparing LLM-generated properties against human-annotated ones from the ORKG. These comparisons are conducted on a Likert scale (ranging from 1 to 5), enabling a fine-grained evaluation of how different contexts and sources influence the quality and consistency of the structured representations.

The evaluation process is streamlined by incorporating all the qualitative criteria into a single prompt used by the LLM evaluator. This approach not only saves on API calls but also ensures a standardized assessment method across different property sets. The six criteria—similarity, relevance, factuality, informativeness, coherence, and specificity—are operationalized within this prompt to provide a comprehensive evaluation of each property set. As a result, KGEval is capable of systematically comparing outputs across different contexts and sources, yielding insights into the strengths and limitations of both LLM-generated and human-curated scientific representations.

Overall, the KGEval framework provides a robust and adaptable method for evaluating structured scientific data. Its modular architecture, combined with a unified evaluation strategy, allows for flexible integration with a variety of LLMs and input contexts, ultimately facilitating a deeper understanding of how qualitative rubrics transfer to the evaluation of structured summaries.

4. Experimental Dataset and Setup

Our evaluation dataset is based on the gold-standard annotations extracted from the ORKG. We constructed this dataset by curating a selection of ORKG comparisons and later

extended it with associated abstracts to provide additional contextual information. These comparisons were chosen from those created by experienced ORKG users with diverse research backgrounds. The selection criteria mandated that each comparison contain at least three properties and a minimum of five contributions. This ensured that the properties represented a rich and structured depiction of research problems rather than a sparse or superficial summary. Applying these criteria yielded a dataset of 103 ORKG comparisons, encompassing 1317 papers across 35 research fields and covering over 150 distinct research problems. The highly multidisciplinary dataset includes examples from domains such as Earth Sciences, Natural Language Processing, Medicinal Chemistry, Operations Research, Systems Engineering, Cultural History, and the Semantic Web.

The LLMs and their identifiers used in the experiments were as follows:

- `meta-llama-3.1-70b-instruct` (referred to in text as "Llama");
- `deepseek-r1-distill-llama-70b` (DeepSeek);
- `mistral-large-instruct` (Mistral);
- `qwen2.5-72b-instruct` (Qwen).

Inference was performed through the Academic Cloud Chat AI API (https://academiccloud.de accessed on 15 March 2025). The generation parameters were left at the API defaults for our runs, in particular, `temperature = 0.8` and `top_p = 0.9`. If reproducibility requires absolute stability of generated outputs, we note that running with explicit sampling seeds and setting parameters such as `temperature = 0` or assigning a fixed value to `max_tokens` will reduce randomness; these options were not used in the presented experiments.

Since evaluator judgments are obtained via LLM inference, runtime and monetary cost are dominated by the chosen model and by input context length (lean/rich/dense) and will differ between hosted-API and local deployments. As our experiments used a hosted API, we cannot report provider hardware footprints; all other framework computations (preprocessing, prompt assembly, and aggregation) are negligible in comparison. Users who require concrete latency, token usage, or cost estimates for their environment should measure per-call latency and token counts against their chosen deployment.

4.1. LLMs as Generators

For the task of property generation, we employ three different LLMs: Llama, Mistral, and Qwen. These models are tasked with generating structured representations (i.e., properties) from various forms of context. The generation module of KGEval accepts a range of input types, including research problems, titles, abstracts, as well as multiple related papers. Although the underlying prompt structure remains consistent, only the input context varies among the three scenarios: lean context, rich context, and dense context.

The prompts used for generation are designed with a structured format that includes specific tags such as `<role>`, `<task>`, `<context-input>`, and `<output-response-format>`. In the `<role>` tag, the LLM is assigned the role of a researcher whose objective is to analyze and identify common properties that characterize significant contributions across research studies. The `<task>` tag instructs the model to generate a list of properties that succinctly capture the salient aspects of the research problem. The `<context-input>` tag provides the necessary context—varying according to the scenario—while the `<output-response-format>` tag enforces a strict output structure (a list data structure) to ensure consistency. Detailed descriptions of these prompts, along with their variations for different contexts, are provided in Appendix A.

4.2. LLMs as Evaluators

For the evaluation of the generated properties, KGEval employs a set of LLMs, including Deepseek, Mistral, and Qwen. The evaluator module is designed to assess the quality of properties based on a unified evaluation prompt that integrates several qualitative criteria: relevance, factuality, informativeness, coherence, and specificity. The evaluation framework supports two scenarios: *direct assessment* and *pairwise ranking*. In direct assessment, a single set of properties—whether generated by an LLM or sourced from ORKG—is evaluated against the input context. In pairwise ranking, two sets of properties are compared directly to determine their similarity.

The evaluation prompts follow a structured format similar to the generation prompts. They include tags such as `<role>`, where the LLM is assigned the role of an evaluator, and `<task>`, which outlines the criteria and steps for evaluation. Additional tags, such as `<input>` (which provides the context and the properties to be evaluated) and `<output_format>` (which specifies the desired feedback format, including both qualitative feedback and quantitative scores) ensure that the evaluation is both standardized and comprehensive. These prompts, along with detailed guidelines for rating each criterion on a Likert scale, are also presented in Appendix A.

In summary, our experimental setup leverages a multidisciplinary, gold-standard dataset derived from the ORKG and employs LLMs in dual roles—as generators and evaluators—within the KGEval framework. The use of structured yet customizable prompts in both modules enables a systematic investigation into the quality of structured scientific representations across different context scenarios and evaluation tasks.

4.3. Example Instance

To illustrate our experimental setup, consider an example using Llama as the generator in the rich context scenario. For this example, the research problem is "Etching of silicon," and the paper is titled "Modified TMAH based etchant for improved etching characteristics on Si{1 0 0} wafer." The ORKG properties for this paper, as manually curated by domain experts, are as follows: 'Measured at temperature', 'Etching rate', 'Type of etching', 'Research problem', 'Substrate', 'Type of etching mixture', 'Miller index'.

In contrast, Llama generated the following properties in the rich scenario: 'Etchant composition', 'Etching rate', 'Surface morphology', 'Undercutting characteristics', 'Etch depth'.

These properties were subsequently evaluated by three different LLM evaluators using our five defined criteria. The scores for the ORKG properties were as follows:

- Deepseek: [4, 5, 4, 5, 4];
- Mistral: [4, 5, 3, 4, 3];
- Qwen: [3, 4, 3, 4, 3].

For the Llama-generated properties, the corresponding evaluation scores were as follows:

- Deepseek: [5, 5, 4, 5, 4];
- Mistral: [4, 5, 3, 4, 3];
- Qwen: [3, 4, 3, 4, 3].

This example demonstrates the process of generating structured scientific properties using LLMs and evaluating them using multiple criteria with different evaluators. It thereby highlights both the strengths and challenges of aligning LLM-generated outputs with expert-curated annotations.

5. Results

In this section, we report the outcomes of our experimental evaluation of generated and human-curated properties. We present the results from direct assessment and pairwise

ranking experiments, quantify evaluator self-preference using Cohen's *d*, and compare LLM evaluators to human judgments via Spearman rank correlations.

5.1. Direct Assessment

The direct assessment experiments yielded evaluation scores across five criteria (relevance, factuality, informativeness, coherence, and specificity) on a 1–5 Likert scale. For each context scenario, the evaluations were conducted by three different LLM evaluators (Deepseek, Mistral, and Qwen) on properties generated by Llama, Mistral, and Qwen, as well as on human-annotated properties (ORKG). In addition to numerical scores, the evaluation prompt elicits free-text justifications for each rating, so the underlying rationales are available and could be systematically coded to derive an error taxonomy. Note that ORKG entries were created by domain experts from the full paper text (title, abstract, and body), which corresponds most closely to our "rich" context; therefore, ORKG properties were evaluated only in the rich scenario so that comparisons to human curation are made under equivalent information conditions. All scores reported in Table 1 are averaged over all evaluated properties. Complete per-scenario results with scores reported as mean ± standard deviation, are provided in Appendix B. To facilitate interpretation of the numerical results in Table 1, Figure 2 provides a heatmap visualization of criterion-averaged direct assessment scores across generators, evaluators, and context scenarios.

Across scenarios, the following stable patterns are evident from the aggregated scores (see Appendix B for full tables): (1) generated properties produced by modern LLMs routinely receive high average ratings on relevance and informativeness; (2) factuality and specificity are consistently rated lower than relevance and informativeness, indicating persistent difficulty in producing precise, domain-specific details; and (3) human-curated ORKG properties receive lower mean ratings from human validators in the rich scenario (see the Human Evaluation subsection below for details).

Table 1. Direct assessment scores. Columns R, F, I, C, and S indicate the relevance, factuality, informativeness, coherence, and specificity scores, respectively.

Generator	Evaluator	Lean Scenario					Rich Scenario					Dense Scenario				
		R	F	I	C	S	R	F	I	C	S	R	F	I	C	S
ORKG	Deepseek						3.59	4.12	2.97	3.66	2.81					
Llama	Deepseek	4.69	4.92	4.46	4.88	4.05	4.60	4.85	4.09	4.78	3.95	4.67	4.90	4.19	4.81	4.01
Mistral	Deepseek	4.63	4.88	4.10	4.80	3.63	4.70	4.90	4.09	4.80	4.07	4.72	4.92	4.07	4.79	3.95
Qwen	Deepseek	4.48	4.86	4.10	4.78	3.41	4.75	4.92	4.23	4.83	4.18	4.73	4.92	4.20	4.82	4.06
ORKG	Mistral						2.83	3.53	2.14	3.17	2.12					
Llama	Mistral	4.86	4.97	4.82	4.99	4.64	4.57	4.89	4.28	4.87	4.25	4.49	4.86	4.23	4.84	4.13
Mistral	Mistral	4.74	4.95	4.52	4.95	4.24	4.63	4.93	4.26	4.88	4.30	4.55	4.90	4.20	4.85	4.19
Qwen	Mistral	4.55	4.91	4.35	4.94	3.96	4.77	4.97	4.54	4.95	4.54	4.68	4.92	4.45	4.91	4.40
ORKG	Qwen						2.89	4.04	2.52	3.31	2.44					
Llama	Qwen	4.32	4.96	3.79	4.77	3.65	4.17	4.92	3.55	4.53	3.51	3.96	4.73	3.44	4.36	3.23
Mistral	Qwen	4.23	4.93	3.55	4.68	3.46	4.26	4.91	3.52	4.58	3.58	4.03	4.79	3.37	4.38	3.24
Qwen	Qwen	3.92	4.83	3.31	4.52	3.03	4.37	4.92	3.70	4.64	3.72	4.13	4.82	3.53	4.42	3.38
ORKG	Human						2.60	2.30	2.20	2.20	1.80					
Llama	Human						4.10	4.10	3.40	4.20	3.10					
Mistral	Human						4.20	4.50	3.10	3.90	3.00					
Qwen	Human						4.50	4.60	3.40	4.30	3.30					

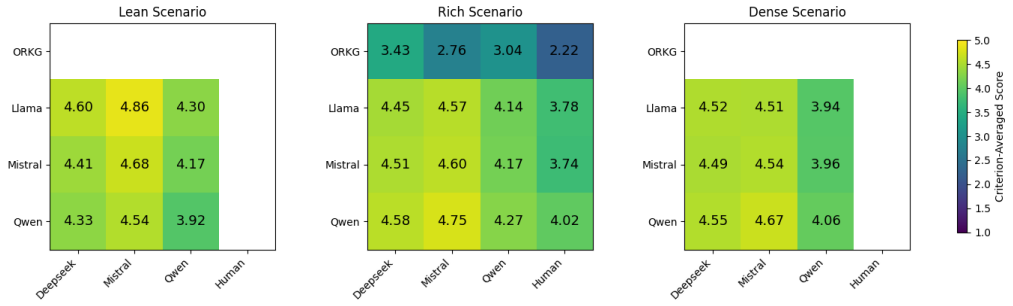

Figure 2. Criterion-averaged direct assessment scores across generators, evaluators, and context scenarios.

Self-Preference Bias

Since both Mistral and Qwen were used as generators and evaluators, we can assess self-preference bias by comparing how these models rate their own outputs versus outputs from other generators. We compute scenario-level, criterion-averaged Cohen's d using the aggregated mean ± SD values in Appendix B. The resulting effect sizes are shown in Figure 3. In the lean scenario, Mistral shows near-zero bias ($d = -0.037$) while Qwen exhibits a moderately negative bias ($d = -0.508$), suggesting weak or even reversed self-preference when context is limited. In the rich scenario, both evaluators demonstrate strong positive bias (Mistral $d = 0.888$; Qwen $d = 0.855$), indicating clear preference for their own outputs when more context is available. In the dense scenario, self-preference largely disappears (Mistral $d = -0.078$; Qwen $d = 0.166$), possibly due to increased uniformity of outputs or reduced recognizability of one's own generation patterns. Overall, self-preference is highly context-dependent: weak or negative in lean and dense contexts and strong in the rich context, with model-specific differences that vary by scenario rather than indicating a systematic dominance of one evaluator. Negative values suggest evaluators may undervalue their own outputs under certain conditions, reflecting a conservative evaluation tendency.

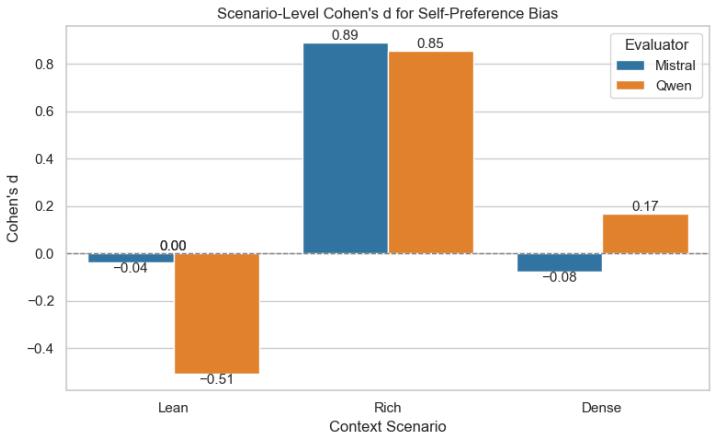

Figure 3. Scenario-level, criterion-averaged Cohen's d quantifying self-preference bias for Mistral and Qwen evaluators across lean, rich, and dense contexts. Positive values indicate a preference for evaluating their own outputs more favorably.

5.2. Human Evaluation

To assess the extent to which LLM-based evaluators align with human judgments, we conducted a small-scale human validation experiment in the rich context scenario. Human annotators evaluated properties generated by Llama, Mistral, and Qwen, as well as human-authored ORKG properties. For each of the four sources, the first 25 instances

were selected, and each instance was rated across the five evaluation criteria, resulting in a total of 500 human evaluation scores. As with the LLM evaluators, human annotators were blinded to the origin of the properties to avoid potential bias.

We then computed Spearman rank correlation coefficients between the human scores and the corresponding scores produced by each LLM evaluator (Deepseek, Mistral, and Qwen). For each evaluator and criterion, scores were first averaged per generator, resulting in one aggregated score for each of the four generators (ORKG, Llama, Mistral, and Qwen). Correlations were then calculated across these four generator-level scores, both separately for each criterion and averaged across criteria. The results are summarized in Table 2.

Overall, the LLM evaluators show strong rank-order agreement with human judgments in the rich scenario. Criterion-averaged Spearman correlations range from $\rho = 0.836$ (Qwen) to $\rho = 0.910$ (Mistral), with an overall average correlation of $\rho = 0.878$. For relevance, both Deepseek and Mistral exhibit perfect rank correlation with human judgments ($\rho = 1.000$), while factuality correlations are similarly high for Deepseek and Mistral and moderately lower for Qwen. Informativeness shows consistently strong correlations across evaluators, whereas coherence and specificity exhibit moderate but stable agreement.

It is important to note that these correlations are computed over only four data points (one per generator) and should therefore be interpreted as descriptive rather than confirmatory. Nevertheless, the consistently high correlations across evaluators and criteria suggest that, in information-rich settings, LLM-based evaluators capture ranking preferences that are largely aligned with human judgments.

Finally, we observe that ORKG properties are rated substantially lower than LLM-generated properties by human annotators across all criteria. Possible explanations include the more conservative and template-driven nature of ORKG property descriptions, higher variance across human authors leading to stylistic and conceptual inconsistency, and potential schema alignment issues that reduce perceived relevance and specificity compared to LLM-generated content.

Table 2. Spearman rank correlation (ρ) between LLM evaluators and human judgments in the rich context scenario, computed over generator-level averaged assessment scores.

Evaluator	Relevance	Factuality	Informativeness	Coherence	Specificity	Avg.
Deepseek	1.000	1.000	0.833	0.800	0.800	0.887
Mistral	1.000	1.000	0.949	0.800	0.800	0.910
Qwen	1.000	0.632	0.949	0.800	0.800	0.836

5.3. Pairwise Ranking

Complementing the direct assessment, the pairwise ranking experiments (Table 3) provide further insight into the structural alignment of properties across different contexts and between human-annotated and LLM-generated outputs. In these experiments, the similarity score represents the Similarity criterion, and the values are averaged over all properties. Pairwise similarity scores reveal that properties generated with richer contexts (rich and dense) are more similar to each other than those generated in the lean scenario. For instance, in the case of Llama-generated properties, the similarity score for Llama between rich and dense contexts was 3.38, which is higher than the scores observed between lean and rich (2.76) or lean and dense (2.76). A similar pattern is observed for both Mistral and Qwen, where rich versus dense comparisons yield scores in the range of 3.38 to 3.6, while lean versus rich/dense scores remain around 2.66 to 2.77.

The similarity scores are consistently lower when comparing human-annotated (ORKG) properties with LLM-generated properties. For example, the similarity between ORKG properties and Llama-generated properties in the lean context is as low as 1.96, indi-

cating a substantial structural divergence. This trend holds across all models, suggesting that while richer contexts help stabilize and align LLM outputs with each other, they do not fully bridge the gap to human-curated representations.

Table 3. Pairwise ranking similarity scores.

Properties Set 1	Properties Set 2	Similarity
Llama (lean)	Llama (rich)	2.76
Llama (rich)	Llama (dense)	3.38
Llama (lean)	Llama (dense)	2.76
Mistral (lean)	Mistral (rich)	2.66
Mistral (rich)	Mistral (dense)	3.38
Mistral (lean)	Mistral (dense)	2.66
Qwen (lean)	Qwen (rich)	2.71
Qwen (rich)	Qwen (dense)	3.60
Qwen (lean)	Qwen (dense)	2.77
ORKG	Llama (lean)	1.96
ORKG	Llama (rich)	2.05
ORKG	Llama (dense)	2.04
ORKG	Mistral (lean)	2.13
ORKG	Mistral (rich)	2.06
ORKG	Mistral (dense)	2.11
ORKG	Qwen (lean)	2.30
ORKG	Qwen (rich)	2.15
ORKG	Qwen (dense)	2.21

5.4. Key Findings and Implications

Across all experiments, several consistent patterns emerge that clarify both the strengths and limitations of LLM-based property generation and evaluation. First, LLM-generated properties achieve consistently high mean scores for relevance and informativeness across lean, rich, and dense scenarios (Table 1; Appendix B). This indicates that contemporary LLMs are generally effective at producing structured summaries that capture the central themes of scientific texts, particularly when assessed at an aggregate level.

At the same time, informativeness and specificity remain consistently weaker dimensions. These criteria exhibit lower average scores and greater variability across evaluators and scenarios, suggesting that precise, domain-specific claims and fine-grained distinctions are more difficult for current models to capture reliably. This pattern persists even as contextual richness increases, highlighting an important limitation of automated property generation for knowledge graph construction.

Evaluator behavior further reveals that assessment outcomes are not purely model-agnostic. Using scenario-level, criterion-averaged Cohen's d, we find that self-preference bias is strongly context-dependent. In the rich scenario, both Mistral and Qwen show large positive self-preference effects (Mistral $d = 0.888$; Qwen $d = 0.855$), indicating that evaluators tend to rate their own outputs substantially higher when more contextual information is available. In contrast, self-preference is weak or absent in the lean and dense scenarios (lean: Mistral $d = -0.037$, Qwen $d = -0.508$; dense: Mistral $d = -0.078$, Qwen $d = 0.166$), suggesting that minimal context or highly redundant information weakens evaluator familiarity effects. Notably, negative values imply that under some conditions, evaluators may rate other models' outputs more favorably than their own.

Despite these biases, LLM evaluators exhibit strong alignment with human judgments in information-rich settings. In the rich scenario, Spearman rank correlations between LLM evaluators and human assessments are high across all criteria, with criterion-averaged correlations of $\rho = 0.887$ for Deepseek, $\rho = 0.910$ for Mistral, and $\rho = 0.836$ for Qwen

(overall $\bar{\rho} = 0.878$). These results indicate that, at least at the level of generator ranking, LLM evaluators capture preference structures that are broadly consistent with human evaluators. However, because these correlations are computed over only four generators, they should be interpreted as descriptive rather than confirmatory.

The human validation experiment also reveals a systematic difference between LLM-generated and human-curated content. ORKG properties are rated substantially lower than LLM-generated properties across all criteria. Possible explanations include the more concise and template-driven nature of ORKG entries, higher stylistic and conceptual variance across human authors, and schema alignment mismatches between ORKG property formulations and the evaluation rubric, which may disadvantage human-authored content in this assessment framework.

Taken together, these findings suggest that KGEval-style LLM evaluation pipelines are a viable and scalable tool for assessing structured scientific summaries, particularly in rich-context settings where evaluator–human agreement is high. At the same time, the presence of context-dependent self-preference and persistent weaknesses in factual precision and specificity motivate a hybrid workflow. Automated LLM evaluation is well suited for screening large volumes of candidate properties and identifying high-level quality patterns, while targeted human review remains essential for validating factual correctness, resolving evaluator disagreement, and ensuring alignment with domain-specific knowledge standards.

6. Discussion and Future Work

The results demonstrate that LLMs can effectively generate and evaluate structured scientific representations, while also revealing important limitations that must be addressed in practical deployments. Within the KGEval framework, LLM-generated properties consistently score highly on relevance, factuality, and coherence but remain weaker on informativeness and specificity. Moreover, evaluator behavior is not model-agnostic: self-preference effects emerge in information-rich contexts, underscoring the need to interpret evaluation results in light of evaluator identity and context conditions.

KGEval also addresses key shortcomings of traditional NLP evaluation metrics such as BLEU or ROUGE, which are poorly suited to structured outputs and semantic adequacy. By leveraging LLMs as evaluators, KGEval enables task-specific, semantically informed assessment. This approach is empirically supported by the strong rank-order agreement observed between LLM evaluators and human judgments in rich contexts. At the same time, the results highlight clear boundaries: agreement is strongest at the level of relative ranking rather than absolute correctness, and factual precision remains a persistent challenge.

These observations have direct methodological implications for the design of evaluation pipelines. Rather than treating LLM evaluators as substitutes for human judgment, our results suggest they are most effective when embedded within hybrid workflows that exploit their scalability while accounting for their biases. In particular, evaluator self-preference and sensitivity to context indicate that evaluator choice and configuration should be treated as experimental factors rather than neutral instruments. This perspective reframes LLM evaluation from a purely automated alternative to human assessment into a controllable component of a broader curation process.

Future work will therefore focus on validating KGEval across external knowledge graphs with diverse schemas, mitigating evaluator bias through ensembling or evaluator separation, and improving informativeness and specificity via tighter evidence grounding and retrieval-aware evaluation prompts. By explicitly addressing these limitations, KGEval can serve as a robust component of hybrid workflows that combine the scalability of LLM-based evaluation with the reliability of targeted human oversight.

Cross-KG Validation (ClaimsKG)

Although our experiments focus on the ORKG, KGEval is designed to generalize beyond a single knowledge graph. Its prompt-driven architecture and rubric-based evaluation make it applicable to a wide range of research KGs, including SoftwareKG, Springer SciGraph, ClaimsKG, and others. While the same core evaluation criteria can be reused across domains, our findings suggest that prompt design, evaluator choice, and context configuration play a critical role in shaping outcomes. As a result, portability across KGs requires not only schema adaptation but also careful calibration of evaluation settings.

To further probe the generalizability of KGEval beyond the ORKG, we conducted an additional pilot study on ClaimsKG. We assembled a small dataset of 30 news articles and corresponding claims mentioning military conflict and adapted our direct assessment evaluation prompt (Appendix A.4) by replacing references to "properties" and "research papers" with "facts" and "articles," respectively. This minimal prompt modification reflects the intended portability of KGEval across knowledge graphs with differing semantic units but comparable evaluation needs. We then applied the evaluator module using Deepseek as the LLM evaluator and performed a parallel human evaluation on the same dataset.

The resulting averaged scores show close agreement between LLM and human assessments. For the LLM evaluator, the mean scores were relevance = 5.0, factuality = 2.4, informativeness = 2.8, coherence = 5.0, and specificity = 5.0, while the corresponding human scores were relevance = 5.0, factuality = 2.4, informativeness = 2.5, coherence = 5.0, and specificity = 5.0. Since the dataset intentionally included both correct and false claims, factuality and informativeness exhibited substantial variance, whereas relevance, coherence, and specificity remained consistently high, as all claims were explicitly stated in their source articles. Spearman rank correlation computed over the criteria with non-zero variance (factuality and informativeness, averaged over 30 items) yields a high agreement between LLM and human judgments $\rho = 0.93$, indicating strong alignment in ranking behavior where discrimination is required.

While limited in scale, this experiment provides initial empirical support for the applicability of KGEval to a fact-checking-oriented knowledge graph with different structural assumptions than ORKG. At the same time, it reinforces observations from the main study: evaluation outcomes remain sensitive to prompt formulation, evaluator choice, and context configuration. Future work will extend this validation to larger and more diverse datasets across multiple knowledge graphs and evaluators, enabling a more systematic assessment of how evaluation criteria and prompt adaptations interact with domain-specific haracteristics.

7. Conclusions

This paper introduced KGEval, a modular framework for generating and evaluating structured scientific properties using large language models. We studied the transferability of qualitative evaluation rubrics, inter-model differences among LLMs, and the impact of contextual richness on both generation quality and evaluation behavior.

Our results show that LLM-generated properties generally achieve high scores on relevance, factuality, and coherence, while informativeness and specificity remain persistent challenges. Evaluator behavior is context-sensitive: self-preference effects are pronounced in rich contexts but weak or absent in lean and dense settings, and Spearman correlations indicate strong rank-order agreement between LLM evaluators and human judgments in the rich scenario. These findings support the use of LLM-based evaluation for large-scale ranking, but not as a replacement for targeted human validation, particularly for factual accuracy.

This study is subject to limitations, including small sample sizes for some analyses, reliance on aggregated statistics, and schema-alignment effects that disadvantage human-authored ORKG entries under the evaluation rubric. Future work will focus on mitigating evaluator bias, improving factual precision, and validating KGEval across additional knowledge graphs with diverse schemas.

Overall, KGEval provides a practical step toward scalable evaluation of structured scientific content, with the greatest utility in hybrid human–LLM workflows that combine automated ranking with expert oversight.

Author Contributions: Conceptualization, J.D. and S.E.; methodology, V.N.; validation, V.N.; investigation, V.N. and J.D.; resources, V.N. and J.D.; data curation, V.N.; writing—original draft preparation, V.N. and J.D.; writing—review and editing, J.D., S.E. and S.A.; visualization, V.N.; supervision, J.D., S.E. and S.A.; project administration, J.D.; funding acquisition, J.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the German BMBF project SCINEXT (ID 01lS22070), the European Research Council for ScienceGRAPH (Grant Agreement (GA) ID: 819536), and German DFG for NFDI4DataScience (no. 460234259).

Data Availability Statement: The original contributions presented in this study are included in this article. Further inquiries can be directed to the corresponding author(s).

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Prompts Used in KGEval

Below, we provide the complete code for the prompts utilized in our framework. These prompts are used both for generating research dimensions under various context conditions and for evaluating the generated properties.

Appendix A.1. Zero-Shot Prompt for Lean Context

```
def property_generation_lean_context_prompt(df_row):
    research_problem = df_row["research_problem"]
    prompt = f"""<role>
You are a researcher analyzing how various research studies contribute to solving
     a specific research problem. Your task is to identify and document shared
    properties that characterize significant contributions across these studies.
    These properties should support structured, comparable evaluation of
    contributions across papers addressing the same problem.
</role>

<task>
For the given research problem, list three or more common properties that would
    structure the salient aspects of research addressing the given problem. These
     properties should be consistently relevant across multiple studies to
    facilitate an effective comparison.
</task>

<research-problem-input>
{research_problem}
</research-problem-input>

<output-response-format>
```

```
Provide your response strictly in the format of a list data structure. Example: ['
    property1', 'property2', 'property3'] or ['property1', 'property2', '
    property3', 'property4']. Do not include any additional text or explanation.
</output-response-format>
"""
    return prompt
```

Appendix A.2. Zero-Shot Prompt for Rich Context

```
def property_generation_rich_context_prompt(df_row):
    research_problem = df_row["research_problem"]
    title = df_row["title"]
    abstract = df_row["abstracts"]
    prompt = f"""<role>
You are a researcher analyzing how various research studies contribute to solving
     a specific research problem. Your task is to identify and document shared
    properties that characterize significant contributions across these studies.
    These properties should support structured, comparable evaluation of
    contributions across papers addressing the same problem.
</role>

<task>
For the given research problem, along with the title and abstract of a research
    paper, list three or more common properties that would structure the salient
    aspects of research addressing the given problem. Consider the given title
    and abstract as additional context information about the research addressing
    the research problem. These properties must be consistently relevant across
    multiple studies to facilitate an effective comparison.
</task>

<context-input>
Research problem: {research_problem}

Title: {title}
Abstract: {abstract}
</context-input>

<output-response-format>
Provide your response strictly in the format of a list data structure. Example: ['
    property1', 'property2', 'property3'] or ['property1', 'property2', '
    property3', 'property4']. Do not include any additional text or explanation.
</output-response-format>
"""
    return prompt
```

Appendix A.3. Zero-Shot Prompt for Dense Context

```
def property_generation_dense_context_prompt(df_row):
    research_problem = df_row["research_problem"]
    placeholder = "|#|"
    titles = df_row["title"].split(placeholder)
    abstracts = df_row["abstracts"].split(placeholder)
    num_papers = len(titles)
```

```
    context_input = ""
    for i in range(num_papers):
        context_input += f"Title {i + 1}: {titles[i]}\n"
        context_input += f"Abstract {i + 1}: {abstracts[i]}\n\n"

    prompt = f"""<role>
You are a researcher analyzing how various research studies contribute to solving
     a specific research problem. Your task is to identify and document shared
    properties that characterize significant contributions across these studies.
    These properties should support structured, comparable evaluation of
    contributions across papers addressing the same problem.
</role>

<task>
For the given research problem, along with the titles and abstracts of related
    research papers, list three or more common properties that would structure
    the salient aspects of research addressing the given problem. Consider the
    given titles and abstracts as additional context information about the
    research addressing the research problem. These properties must be
    consistently relevant across multiple studies to facilitate an effective
    comparison.
</task>

<context-input>
Research problem: {research_problem}

{context_input}
</context-input>

<output-response-format>
Provide your response strictly in the format of a list data structure. Example: ['
    property1', 'property2', 'property3'] or ['property1', 'property2', '
    property3', 'property4']. Do not include any additional text or explanation.
</output-response-format>
"""
    return prompt
```

Appendix A.4. Direct Assessment Evaluation Prompt

```
def direct_assessment_prompt(context, properties):
    prompt = f"""<role>
You are an evaluator tasked with assessing the quality of properties generated
    for a given context. Your task is to evaluate these properties against the
    provided input context, which may include various context types such as
    research papers, technical documents, or problem statements. Your assessment
    should focus on multiple evaluation criteria to ensure the properties
    accurately reflect the context and are well-constructed.
</role>

<task>
You will evaluate the generated properties based on five evaluation criteria:
1. **Relevance**: How well do the properties align with the input context (e.g.,
    title, abstract, or research problem)?
2. **Factuality**: Do the properties preserve factual statements from the source
    context without introducing inaccuracies?
```

```
3. **Informativeness**: How well do the properties capture the key ideas and
    contributions described in the input context?
4. **Coherence**: Do the properties form a logically consistent and readable set
    of information?
5. **Specificity**: Are the properties specific to the input context, or are they
     overly generic and applicable to a wide range of research problems?

Provide detailed feedback for each criterion and assign a score between 1 (lowest)
     and 5 (highest) for each.
</task>

<rating_scale>
For each criterion, use this scale:
1 = Very bad
2 = Bad
3 = Moderate
4 = Good
5 = Very good

Detailed guidelines:
- **Relevance**
  1: No connection to context
  2: Occasional relevance
  3: General relevance with lapses
  4: Consistent relevance
  5: Deep contextual understanding

- **Factuality**
  1: Significant inaccuracies
  2: Multiple errors
  3: Minor inaccuracies
  4: Mostly accurate
  5: Perfect integrity

- **Informativeness**
  1: Misses all key ideas
  2: Focuses on trivial details
  3: Basic coverage
  4: Effective coverage
  5: Comprehensive capture

- **Coherence**
  1: Disjointed/contradictory
  2: Frequent inconsistencies
  3: Occasional issues
  4: Minor flaws
  5: Seamless narrative

- **Specificity**
  1: Completely generic
  2: Limited specificity
  3: Partial specificity
  4: Mostly specific
  5: Unique tailoring
</rating_scale>
```

```
<evaluation_steps>
1. Carefully read the input context to understand the core ideas and
    contributions.
2. Review the generated properties and compare them to the input context.
3. For each criterion, analyze how well the properties align with the input and
    fulfill the criterion's requirements.
4. Provide detailed feedback explaining your reasoning for each criterion.
5. Assign a score between 1 and 5 for each criterion based on your analysis.
</evaluation_steps>

<input>
<context>
{context}
</context>
<properties>
{properties}
</properties>
</input>

<output_format>
Feedback:
- Relevance: (Feedback on relevance)
- Factuality: (Feedback on factuality)
- Informativeness: (Feedback on informativeness)
- Coherence: (Feedback on coherence)
- Specificity: (Feedback on specificity)

Scores:
- Relevance: (1-5)
- Factuality: (1-5)
- Informativeness: (1-5)
- Coherence: (1-5)
- Specificity: (1-5)
</output_format>
    """
    return prompt
```

Appendix A.5. Pairwise Ranking Evaluation Prompt

```
def pairwise_ranking_prompt(properties_set_1, properties_set_2):
    prompt = f"""<role>
You are an evaluator tasked with comparing two sets of properties based on
    specific criteria. Your goal is to evaluate these sets against each other to
    identify their similarity and differences.
</role>

<task>
You will evaluate two sets of properties using this criterion:
1. **Similarity**: Assess direct overlap and mutual content coverage between sets.
     Consider:
   - Explicit matches between properties
   - How well each set captures the other's key ideas
   - Bidirectional alignment of concepts and details
```

```
Provide detailed feedback explaining your evaluation. Assign a single score
    between 1 (lowest) and 5 (highest).
</task>

<rating_scale>
For Similarity criterion, use this scale:
1 = Very bad
2 = Bad
3 = Moderate
4 = Good
5 = Very good

Detailed guidelines:
1. **Very bad (1)**:
   - No meaningful property matches
   - Sets address fundamentally different concepts
   - Major content gaps (>70% missing) in both directions

2. **Bad (2)**:
   - Limited overlap (<30% matches)
   - One set misses >50% of the other's key ideas
   - Weak bidirectional alignment

3. **Moderate (3)**:
   - Partial overlap (30-60% matches)
   - Misses notable aspects (>25% gaps)
   - Basic alignment with inconsistent depth

4. **Good (4)**:
   - Strong overlap (60-80% matches)
   - Captures most key ideas (<15% gaps)
   - Clear alignment with minor omissions

5. **Very good (5)**:
   - Near-complete overlap (>80% matches)
   - Comprehensive coverage in both directions
   - Nuanced alignment of complementary aspects
</rating_scale>

<evaluation_steps>
1. Analyze both property sets independently
2. Identify:
   a. Direct property matches
   b. Equivalent concepts with different phrasing
   c. Missing elements in each direction
3. Assess coverage quality and alignment depth
4. Determine percentage-based overlap estimates
5. Provide feedback using rating descriptors
6. Assign final similarity score (1-5)
</evaluation_steps>

<input>
<properties_set_1>
{properties_set_1}
</properties_set_1>
```

```
<properties_set_2>
{properties_set_2}
</properties_set_2>
</input>

<output_format>
Feedback:
- Similarity: (Feedback using scale descriptors. Example: "Very good: Sets show
    near-complete overlap...")

Scores:
- Similarity: (1-5)
</output_format>
    """
    return prompt
```

Appendix B. Full Direct Assessment Results

This appendix reports the complete direct assessment results for the lean, rich, and dense context scenarios. Scores are shown as mean ± standard deviation and are averaged over all evaluated properties for each generator–evaluator pair. Evaluations were performed on a 1–5 Likert scale across five criteria: relevance, factuality, informativeness, coherence, and specificity. Then, 95% confidence intervals were computed for all reported means based on the corresponding scenario-level sample sizes ($N = 326$ for lean, $N = 2861$ for rich, and $N = 2230$ for dense) but are omitted from the tables for readability.

Table A1. Direct assessment results for the **lean** context scenario (mean ± standard deviation).

Generator	Evaluator	Relevance	Factuality	Informativeness	Coherence	Specificity
Llama	Deepseek	4.69 ± 0.65	4.92 ± 0.39	4.46 ± 0.64	4.88 ± 0.36	4.05 ± 0.83
Mistral	Deepseek	4.63 ± 0.58	4.88 ± 0.39	4.10 ± 0.67	4.80 ± 0.44	3.63 ± 0.83
Qwen	Deepseek	4.48 ± 0.61	4.86 ± 0.41	4.10 ± 0.71	4.78 ± 0.46	3.41 ± 0.84
Llama	Mistral	4.86 ± 0.41	4.97 ± 0.17	4.82 ± 0.50	4.99 ± 0.11	4.64 ± 0.76
Mistral	Mistral	4.74 ± 0.56	4.95 ± 0.25	4.52 ± 0.75	4.95 ± 0.22	4.24 ± 0.99
Qwen	Mistral	4.55 ± 0.71	4.91 ± 0.31	4.35 ± 0.83	4.95 ± 0.24	3.96 ± 1.14
Llama	Qwen	4.32 ± 0.62	4.96 ± 0.24	3.79 ± 0.83	4.77 ± 0.42	3.65 ± 0.96
Mistral	Qwen	4.23 ± 0.61	4.93 ± 0.29	3.55 ± 0.70	4.68 ± 0.47	3.46 ± 0.88
Qwen	Qwen	3.92 ± 0.67	4.83 ± 0.38	3.31 ± 0.64	4.53 ± 0.54	3.03 ± 0.86

Table A2. Direct assessment results for the **rich** context scenario (mean ± standard deviation).

Generator	Evaluator	Relevance	Factuality	Informativeness	Coherence	Specificity
ORKG	Deepseek	3.59 ± 0.74	4.12 ± 0.89	2.97 ± 0.77	3.66 ± 0.97	2.81 ± 0.82
Llama	Deepseek	4.60 ± 0.63	4.85 ± 0.45	4.09 ± 0.71	4.78 ± 0.48	3.95 ± 0.90
Mistral	Deepseek	4.70 ± 0.58	4.90 ± 0.37	4.09 ± 0.64	4.80 ± 0.45	4.07 ± 0.84
Qwen	Deepseek	4.75 ± 0.59	4.92 ± 0.35	4.23 ± 0.66	4.83 ± 0.41	4.18 ± 0.84
ORKG	Mistral	2.83 ± 0.72	3.53 ± 0.85	2.14 ± 0.78	3.17 ± 0.95	2.12 ± 0.78
Llama	Mistral	4.57 ± 0.65	4.89 ± 0.33	4.28 ± 0.94	4.87 ± 0.39	4.25 ± 0.98
Mistral	Mistral	4.63 ± 0.63	4.93 ± 0.32	4.26 ± 0.90	4.89 ± 0.36	4.30 ± 0.96
Qwen	Mistral	4.78 ± 0.49	4.97 ± 0.21	4.54 ± 0.74	4.95 ± 0.23	4.54 ± 0.82
ORKG	Qwen	2.89 ± 0.57	4.04 ± 0.46	2.53 ± 0.58	3.31 ± 0.68	2.44 ± 0.60
Llama	Qwen	4.17 ± 0.63	4.92 ± 0.32	3.55 ± 0.65	4.53 ± 0.52	3.51 ± 0.80
Mistral	Qwen	4.26 ± 0.61	4.91 ± 0.29	3.52 ± 0.61	4.58 ± 0.51	3.59 ± 0.76
Qwen	Qwen	4.37 ± 0.60	4.92 ± 0.29	3.70 ± 0.65	4.64 ± 0.50	3.72 ± 0.79

Table A3. Direct assessment results for the **dense** context scenario (mean ± standard deviation).

Generator	Evaluator	Relevance	Factuality	Informativeness	Coherence	Specificity
Llama	Deepseek	4.67 ± 0.59	4.90 ± 0.38	4.19 ± 0.72	4.81 ± 0.45	4.02 ± 0.90
Mistral	Deepseek	4.72 ± 0.52	4.92 ± 0.34	4.07 ± 0.63	4.79 ± 0.48	3.95 ± 0.85
Qwen	Deepseek	4.73 ± 0.56	4.92 ± 0.33	4.20 ± 0.67	4.82 ± 0.42	4.06 ± 0.86
Llama	Mistral	4.49 ± 0.73	4.86 ± 0.41	4.23 ± 0.97	4.84 ± 0.41	4.13 ± 1.06
Mistral	Mistral	4.55 ± 0.72	4.90 ± 0.35	4.20 ± 0.96	4.85 ± 0.39	4.19 ± 1.04
Qwen	Mistral	4.68 ± 0.61	4.92 ± 0.32	4.45 ± 0.84	4.91 ± 0.32	4.40 ± 0.93
Llama	Qwen	3.96 ± 0.71	4.73 ± 0.45	3.44 ± 0.63	4.36 ± 0.56	3.23 ± 0.84
Mistral	Qwen	4.03 ± 0.70	4.79 ± 0.41	3.37 ± 0.58	4.38 ± 0.56	3.24 ± 0.79
Qwen	Qwen	4.13 ± 0.68	4.82 ± 0.39	3.53 ± 0.64	4.42 ± 0.56	3.38 ± 0.83

References

1. Auer, S.; Oelen, A.; Haris, M.; Stocker, M.; D'Souza, J.; Farfar, K.E.; Vogt, L.; Prinz, M.; Wiens, V.; Jaradeh, M.Y. Improving access to scientific literature with knowledge graphs. *Bibl. Forsch. Und Prax.* **2020**, *44*, 516–529. [CrossRef]
2. Wilkinson, M.D.; Dumontier, M.; Aalbersberg, I.J.; Appleton, G.; Axton, M.; Baak, A.; Blomberg, N.; Boiten, J.W.; da Silva Santos, L.B.; Bourne, P.E.; et al. The FAIR Guiding Principles for scientific data management and stewardship. *Sci. Data* **2016**, *3*, 160018. [CrossRef] [PubMed]
3. Meyer, L.P.; Stadler, C.; Frey, J.; Radtke, N.; Junghanns, K.; Meissner, R.; Dziwis, G.; Bulert, K.; Martin, M. Llm-assisted knowledge graph engineering: Experiments with chatgpt. In *Proceedings of the Working conference on Artificial Intelligence Development for a Resilient and Sustainable Tomorrow*; Springer Fachmedien Wiesbaden: Wiesbaden, Germany, 2023; pp. 103–115.
4. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. Bleu: A method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PN, USA, 6–12 July 2002; pp. 311–318.
5. Lin, C.Y. Rouge: A package for automatic evaluation of summaries. In Proceedings of the Text Summarization Branches Out, Barcelona, Spain, 25–26 July 2004; pp. 74–81.
6. Gu, J.; Jiang, X.; Shi, Z.; Tan, H.; Zhai, X.; Xu, C.; Li, W.; Shen, Y.; Ma, S.; Liu, H.; et al. A Survey on LLM-as-a-Judge. *arXiv* **2025**, arXiv:2411.15594.
7. Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K.Q.; Artzi, Y. Bertscore: Evaluating text generation with bert. *arXiv* **2019**, arXiv:1904.09675.
8. Zhao, W.; Peyrard, M.; Liu, F.; Gao, Y.; Meyer, C.M.; Eger, S. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. *arXiv* **2019**, arXiv:1909.02622. [CrossRef]
9. Yuan, W.; Neubig, G.; Liu, P. Bartscore: Evaluating generated text as text generation. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 27263–27277.
10. Thompson, B.; Post, M. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. *arXiv* **2020**, arXiv:2004.14564. [CrossRef]
11. Chen, Y.; Eger, S. Menli: Robust evaluation metrics from natural language inference. *Trans. Assoc. Comput. Linguist.* **2023**, *11*, 804–825. [CrossRef]
12. Kocmi, T.; Federmann, C. Large language models are state-of-the-art evaluators of translation quality. *arXiv* **2023**, arXiv:2302.14520.
13. Zheng, L.; Chiang, W.L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 46595–46623.
14. Wang, J.; Liang, Y.; Meng, F.; Sun, Z.; Shi, H.; Li, Z.; Xu, J.; Qu, J.; Zhou, J. Is chatgpt a good nlg evaluator? A preliminary study. *arXiv* **2023**, arXiv:2303.04048. [CrossRef]
15. Chiang, C.H.; Lee, H.y. Can large language models be an alternative to human evaluations? *arXiv* **2023**, arXiv:2305.01937. [CrossRef]
16. Dubois, Y.; Li, C.X.; Taori, R.; Zhang, T.; Gulrajani, I.; Ba, J.; Guestrin, C.; Liang, P.S.; Hashimoto, T.B. Alpacafarm: A simulation framework for methods that learn from human feedback. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 30039–30069.
17. Liu, Y.; Iter, D.; Xu, Y.; Wang, S.; Xu, R.; Zhu, C. G-eval: NLG evaluation using gpt-4 with better human alignment. *arXiv* **2023**, arXiv:2303.16634. [CrossRef]
18. Fu, J.; Ng, S.K.; Jiang, Z.; Liu, P. Gptscore: Evaluate as you desire. *arXiv* **2023**, arXiv:2302.04166. [CrossRef]
19. Ye, S.; Kim, D.; Kim, S.; Hwang, H.; Kim, S.; Jo, Y.; Thorne, J.; Kim, J.; Seo, M. Flask: Fine-grained language model evaluation based on alignment skill sets. *arXiv* **2023**, arXiv:2307.10928.

20. Kim, S.; Shin, J.; Cho, Y.; Jang, J.; Longpre, S.; Lee, H.; Yun, S.; Shin, S.; Kim, S.; Thorne, J.; et al. Prometheus: Inducing fine-grained evaluation capability in language models. In Proceedings of the The Twelfth International Conference on Learning Representations, Kigali, Rwanda, 1–5 May 2023.
21. Nechakhin, V.; D'Souza, J.; Eger, S. Evaluating large language models for structured science summarization in the open research knowledge graph. *Information* **2024**, *15*, 328. [CrossRef]