

Article

GraderAssist: A Graph-Based Multi-LLM Framework for Transparent and Reproducible Automated Evaluation

Catalin Anghel ^{1,*}, Andreea Alexandra Anghel ², Emilia Pecheanu ¹, Adina Cocu ¹,
Marian Viorel Craciun ^{1,*}, Paul Iacobescu ³, Antonio Stefan Balau ² and Constantin Adrian Andrei ⁴

- ¹ Department of Computer Science and Information Technology, “Dunărea de Jos” University of Galati, Științei St. 2, 800201 Galati, Romania; emilia.pecheanu@ugal.ro (E.P.); adina.cocu@ugal.ro (A.C.)
- ² Computer Science and Information Technology Program, Faculty of Automation, Computer Science, Electrical and Electronic Engineering, “Dunărea de Jos” University of Galati, 800201 Galati, Romania; aa481@student.ugal.ro (A.A.A.); ab826@student.ugal.ro (A.S.B.)
- ³ Doctoral School, “Dunărea de Jos” University of Galati, 800201 Galati, Romania; paul.iacobescu@ugal.ro
- ⁴ “Foisor” Clinical Hospital of Orthopaedics, Traumatology and Osteoarticular TB, 021382 Bucharest, Romania; constantin_andrei@drd.umfcd.ro
- * Correspondence: catalin.anghel@ugal.ro (C.A.); marian.craciun@ugal.ro (M.V.C.)

Abstract

Background and objectives: Automated evaluation of open-ended responses remains a persistent challenge, particularly when consistency, transparency, and reproducibility are required. While large language models (LLMs) have shown promise in rubric-based evaluation, their reliability across multiple evaluators is still uncertain. Variability in scoring, feedback, and rubric adherence raises concerns about interpretability and system robustness. This study introduces GraderAssist, a graph-based, rubric-guided, multi-LLM framework designed to ensure transparent and reproducible automated evaluation. **Methods:** GraderAssist evaluates a dataset of 220 responses to both technical and argumentative questions, collected from undergraduate computer science courses. Six open-source LLMs and GPT-4 (as expert reference) independently scored each response using two predefined rubrics. All outputs—including scores, feedback, and metadata—were parsed, validated, and stored in a Neo4j graph database, enabling structured querying, traceability, and longitudinal analysis. **Results:** Cross-model analysis revealed systematic differences in scoring behavior and feedback generation. Some models produced more generous evaluations, while others aligned closely with GPT-4. Semantic analysis using Sentence-BERT embeddings highlighted distinctive feedback styles and variable rubric adherence. Inter-model agreement was stronger for technical criteria but diverged substantially for argumentative tasks. **Originality:** GraderAssist integrates rubric-guided evaluation, multi-model comparison, and graph-based storage into a unified pipeline. By emphasizing reproducibility, transparency, and fine-grained analysis of evaluator behavior, it advances the design of interpretable automated evaluation systems with applications in education and beyond.

Keywords: GraderAssist; automated assessment; large language models; rubric-based evaluation; assessment consistency; assessment fairness



Academic Editor: Antony Bryant

Received: 23 September 2025

Revised: 1 November 2025

Accepted: 6 November 2025

Published: 9 November 2025

Citation: Anghel, C.; Anghel, A.A.; Pecheanu, E.; Cocu, A.; Craciun, M.V.; Iacobescu, P.; Balau, A.S.; Andrei, C.A. GraderAssist: A Graph-Based Multi-LLM Framework for Transparent and Reproducible Automated Evaluation. *Informatics* **2025**, *12*, 123. <https://doi.org/10.3390/informatics12040123>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. Background and Motivation

Assessment in higher education, particularly in STEM disciplines, requires evaluating not only factual knowledge but also practical application, reasoning processes, and communication skills [1]. While principles such as validity, reliability, fairness, and transparency

are essential, their consistent application is difficult in large cohorts and with open-ended responses [2]. In our study, we specifically targeted these challenges by operationalizing two rubrics: a technical rubric (accuracy, clarity, completeness, terminology) and an argumentative rubric (clarity, coherence, originality, dialecticality). These dimensions capture the precision of technical knowledge, the applicability of concepts to practical contexts, and the depth of reasoning in argumentative tasks, providing a structured foundation for automated evaluation [3].

Automated evaluation of open-ended student responses presents persistent challenges in higher education [4]. Unlike objective questions, open-format responses require multidimensional judgment aligned with pedagogical rubrics [3]. These include content-specific accuracy, structural clarity, completeness of reasoning, and appropriate use of terminology [5]. Manual evaluation along such dimensions is time-consuming, inconsistent across graders, and often lacks transparency for students [2].

Instruction-tuned large language models (LLMs) have demonstrated strong potential for automating aspects of grading in educational contexts [6]. These models are fine-tuned on datasets containing explicit task instructions and human feedback, enabling them to follow evaluation prompts more reliably than base models [7]. Experimental results show that models such as GPT-4o produce scores highly correlated with those assigned by human graders, with Spearman rank correlation (ρ) values exceeding 0.96 on short-answer and code-based tasks [8]. Spearman's ρ is a non-parametric statistic measuring the strength and direction of a monotonic relationship between two ranked variables [9]. Equivalence testing confirms that the scoring reliability of premium LLMs approaches human-level variance under rubric-based supervision [8]. Additional studies report consistent LLM-human alignment in tasks involving language generation, concept classification, and formative assessment [6].

Despite these advances, rubric-aligned evaluation across multiple dimensions remains insufficiently explored. Most existing systems focus on global correctness or concept detection, neglecting finer-grained criteria. Furthermore, little is known about how consistently different LLMs apply the same evaluation rubric, and key aspects such as rubric interpretation, feedback generation, and judgment style remain poorly understood across models. Related experiments in technical domains such as physics confirm that LLMs can be adapted for structured problem grading, but do not investigate inter-model consistency or feedback alignment [10]. These limitations constrain the reliability, interpretability, and pedagogical utility of LLM-based assessment. Structured AI-based decision support has shown effectiveness in other high-stakes domains such as medical diagnostics [11–13], motivating similar approaches in educational contexts.

These challenges emphasize the importance of a systematic and transparent framework to ensure fairness, consistency, and traceability in automated educational assessment. The rubric-guided multi-LLM framework introduced in this paper serves as the foundation for an AI assistant designed to evaluate assignments, tests, and other educational tasks. Rather than replacing human evaluators, it supports them by providing assistance, reducing workload, saving time, and making grading more efficient, reliable, and scalable—especially when handling large volumes of content.

1.2. Research Gap, Objectives, and Contributions

Although large language models (LLMs) have demonstrated impressive capabilities in generating and evaluating natural language content, their application to structured, rubric-aligned educational assessment remains limited. Most prior studies have concentrated on single-model evaluation or correctness-oriented scoring, without incorporating multidimensional rubrics or investigating evaluator behavior across multiple models [14,15].

Only a limited number of approaches address fine-grained assessment criteria such as completeness, terminology, coherence, or dialectical engagement, and even fewer examine how different LLMs interpret and apply these criteria to the same student responses [16]. Research on feedback generation often emphasizes fluency and perceived helpfulness, while overlooking essential aspects such as consistency, semantic divergence, and explicit alignment with rubric standards [6,16].

Most existing automated evaluation frameworks rely on flat storage formats such as CSV or JSON, which fail to provide structural traceability between student answers, rubric criteria, scoring models, and feedback explanations [17]. This limitation constrains the possibility of conducting longitudinal analyses, comparing evaluators across tasks, and verifying model behavior over time [18]. To achieve interpretable and reproducible assessment, a persistent and structured representation is required—one that supports evaluation traceability and criterion-level querying, particularly when multiple LLMs operate independently under the same educational rubric.

This study addresses these challenges by introducing GraderAssist, a rubric-guided framework for evaluating open-ended student responses that integrates rubric-based scoring, multi-model evaluation, and structured storage of outputs. GraderAssist applies predefined technical and argumentative rubrics, each containing four criteria, and compares the outputs of six instruction-tuned LLMs with GPT-4 as an expert reference. Each evaluation instance—comprising the student answer, model-assigned scores, and qualitative feedback—is stored in a Neo4j graph database, enabling structured querying, inter-model comparison, and longitudinal analysis.

GraderAssist enables both global and fine-grained comparisons of evaluation results. In particular, it supports criterion-level analysis by tracking how each model scores individual rubric dimensions (e.g., clarity, accuracy, completeness, or dialecticality), and by enabling detailed inspection of rubric adherence in the generated feedback. These capabilities are essential for understanding evaluator behavior, detecting judgment variation, and improving the fairness and consistency of automated assessment.

The combination of multi-model evaluation, multidimensional rubrics, and graph-based storage provides criterion-level provenance and transparent, queryable traces absent from prior single-model or flat-file approaches.

This study makes the following key contributions:

- It introduces GraderAssist, a rubric-guided multi-LLM framework for evaluating open-ended tasks;
- It formalizes a multi-evaluator setting with several LLMs as independent graders and GPT-4 as a reference for consistency analyses;
- It profiles model feedback (embeddings and clustering) to compare style, rubric adherence, and evaluative depth;
- It provides graph-based storage for traceable, reproducible, and queryable evaluation;
- It demonstrates the pedagogical relevance of rubric-guided multi-model assessment, emphasizing fairness, reliability, and interpretability.

1.3. Related Work

The use of large language models for evaluating natural language output has gained traction across multiple domains, including education, dialogue systems, and summarization. Several frameworks have explored the potential of LLMs to act as evaluators, either through direct scoring or pairwise comparison. Early work in LLM-based judgment focused on fluency, coherence, and relevance in dialogue contexts, as demonstrated by systems like ChatEval [19] and Arena [20], where LLMs assessed response quality across open-ended prompts.

More recent approaches have applied rubric-based scoring to educational tasks. RubricEval [21] introduced a scalable method for grading open-ended student responses using predefined criteria and human-model comparisons, although it focused exclusively on single-model evaluation and lacked infrastructure for traceability. Similarly, LLM-Rubric investigated multidimensional scoring for natural language outputs, showing that fine-tuned models can reliably score multiple dimensions such as clarity or originality, but did not assess inter-model variability [22].

Beyond education, other evaluation pipelines have explored iterative or multi-agent methods for improving scoring quality. TEaR introduced a mechanism for self-refinement by allowing LLMs to reevaluate and improve their own outputs through systematic feedback loops [23]. Reflexion [24] applied a similar idea with verbal reinforcement and internal critique. Auto-Arena [25] extended this paradigm by orchestrating multi-agent peer battles and committee discussions for robust comparative evaluation.

While these systems bring valuable techniques for enhancing LLM judgments, they primarily target general generative or translation tasks, not rubric-aligned educational assessment [23,25]. Moreover, most existing evaluation frameworks rely on flat output storage, which restricts their ability to trace evaluator behavior, analyze feedback patterns, and support criterion-level inspection [17].

2. Materials and Methods

2.1. Dataset and Rubric Design

The dataset comprised 220 open-ended responses collected from undergraduate students enrolled in the Computer Science and Information Technology program at the Faculty of Automation, Computers, Electrical Engineering and Electronics, “Dunărea de Jos” University of Galati (Romania). The data were collected through an online survey form administered as part of a formative academic activity. Students provided written answers to 10 diverse questions—five technical and five argumentative—designed to elicit explanations, reasoning, and content-specific knowledge. Participation was anonymous, and all respondents gave informed consent for the use of their answers in a scientific study. The final item on the form explicitly asked: “Are you willing to allow your anonymous answers to be used for scientific research?”, with binary options (Yes/No). Only affirmative responses were retained for analysis.

Each data instance was structured as a tuple containing the pseudonymized *student_id*, *question_id*, *rubric_type* (technical or argumentative), and *answer_text*. No personal data were collected.

Two predefined rubrics were employed to guide the evaluation process, one for technical questions and one for argumentative questions. Each rubric comprised four specific criteria, explicitly defined to reflect the nature of the corresponding question type [5,26]. The technical rubric included accuracy (correctness of statements and solution steps relative to the task requirements), clarity (readability of explanations and structure of the answer), completeness (coverage of all required parts of the prompt and inclusion of necessary steps or justifications), and terminology (appropriate and consistent use of domain-specific terms and notation). The argumentative rubric evaluated clarity (clear thesis and understandable phrasing), coherence (logical organization and consistent linkage between claims, evidence, and conclusions), originality (presence of non-trivial ideas or perspectives beyond paraphrase), and dialecticality (engagement with plausible counterarguments and appropriate rebuttals). All criteria were scored independently on a 10-point scale to ensure comparability across evaluators. Rubric definitions were stored in structured JSON format (technical.json and argumentative.json) to ensure consistent interpretation across LLM evaluators and to enable criterion-specific analysis of scores and feedback [27].

To ensure transparency and reproducibility, Table 1 lists the full set of questions used in this study. The questions were designed to balance factual knowledge with open-ended reasoning, reflecting typical topics in computer science education and supporting the development of both technical and argumentative competencies.

Table 1. Full list of student questions, categorized by rubric type.

Question ID	Rubric Type	Question Text
Q1	Technical	What is a version control system and why is it important in software development?
Q2	Technical	Explain the difference between narrow AI and general AI. Provide examples for each.
Q3	Technical	What is the Naive Bayes classifier and how is it applied in decision-making?
Q4	Technical	Describe the main steps of the search algorithm used in an expert system.
Q5	Technical	What role does facial recognition play in modern security applications?
Q6	Argumentative	Should the use of facial recognition systems be restricted in public spaces? Justify your answer.
Q7	Argumentative	Should teachers accept the use of GitHub by students instead of uploading assignments to Moodle?
Q8	Argumentative	Can an AI-based system make ethical decisions responsibly? Argue.
Q9	Argumentative	To what extent do online personalization algorithms affect users' autonomy?
Q10	Argumentative	Is it correct to consider that a machine that passes the Turing Test is "intelligent"? Why or why not?

2.2. Evaluation Pipeline and Prompt Structure

The evaluation framework followed a modular, five-stage pipeline designed to ensure consistency, traceability, and scalable rubric-based assessment. The system was built to operate across multiple instruction-tuned LLMs, using standardized prompts and structured output to enable reliable comparison. The high-level logic of the rubric-guided evaluation process is summarized in Figure 1. This schematic illustrates the core components: starting from raw student input, the system selects the appropriate rubric, generates a structured prompt, dispatches the prompt to multiple models, and stores the resulting evaluations in a graph-based database.

Each evaluation instance began with a pair consisting of a question and an open-ended student response—that is, a written answer requiring free-form explanation or argumentation, as opposed to multiple-choice or binary formats. Additional metadata—such as the anonymized student ID, question ID, and rubric type—was also included. Based on the rubric type (technical or argumentative), the rubric selector assigned one of two predefined JSON-based rubrics. Each rubric contained four explicitly defined criteria, designed to reflect either content mastery or argumentation skills.

The evaluation rubric was selected according to the question type. Technical questions were assessed in terms of factual accuracy and terminological precision, while argumentative questions emphasized reasoning structure and dialectical engagement. Table 2 lists the specific criteria used for each rubric.

Once the rubric was selected, the prompt generator constructed a structured instruction. This prompt included (1) an expert-level role definition for the evaluator; (2) a list of rubric criteria, each presented with its name and definition in a clear, itemized format; (3) the question and answer; and (4) a strict requirement to return output in a JSON-only format. This structure ensured compatibility across models and simplified automated parsing.

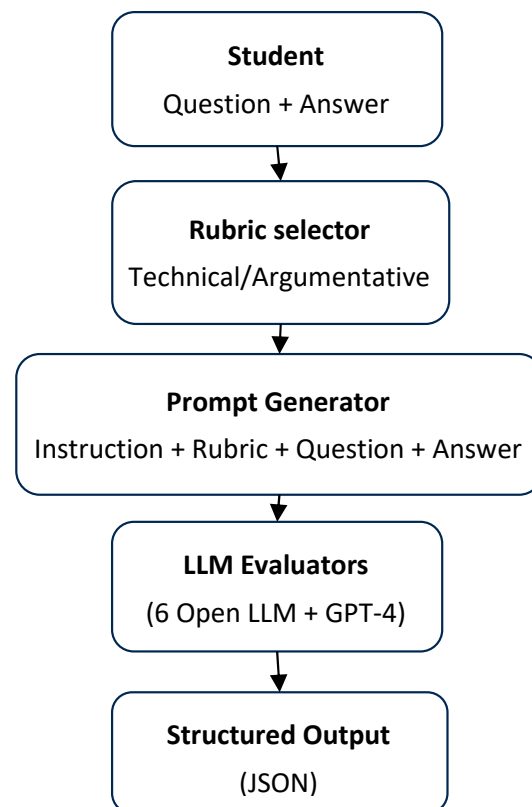


Figure 1. Rubric-based evaluation pipeline using multiple LLMs and structured prompt generation.

Table 2. Rubric criteria for technical and argumentative questions.

Rubric Type	Criterion 1	Criterion 2	Criterion 3	Criterion 4
Technical	Accuracy	Clarity	Completeness	Terminology
Argumentative	Clarity	Coherence	Originality	Dialecticality

The prompt was then passed to each configured model in the evaluation suite, which included six open-source LLMs (e.g., LLaMA3, Nous-Hermes2, Dolphin-Mistral) and GPT-4 as reference. Each model processed the input independently and generated scores from 1 to 10 for each rubric criterion, along with qualitative feedback.

In the fourth stage, the system extracted and validated the JSON output using robust parsing functions. It detected malformed content, applied fallback strategies if necessary, and normalized the feedback structure for storage.

Finally, the validated evaluation was saved to a Neo4j graph database. All relevant entities—Student, Question, Answer, Model, and Evaluation—were recorded as interconnected nodes, supporting full traceability and longitudinal analysis.

A more detailed breakdown of the five-stage pipeline is presented in Figure 2, which outlines the input, operation, and output of each step in a structured format.

This modular pipeline ensured that all student responses were evaluated by large language models (LLMs) under identical rubric definitions and standardized prompts. As a result, evaluations were conducted in a consistent, interpretable, and traceable manner, supporting fairness and reproducibility regardless of the specific model used.

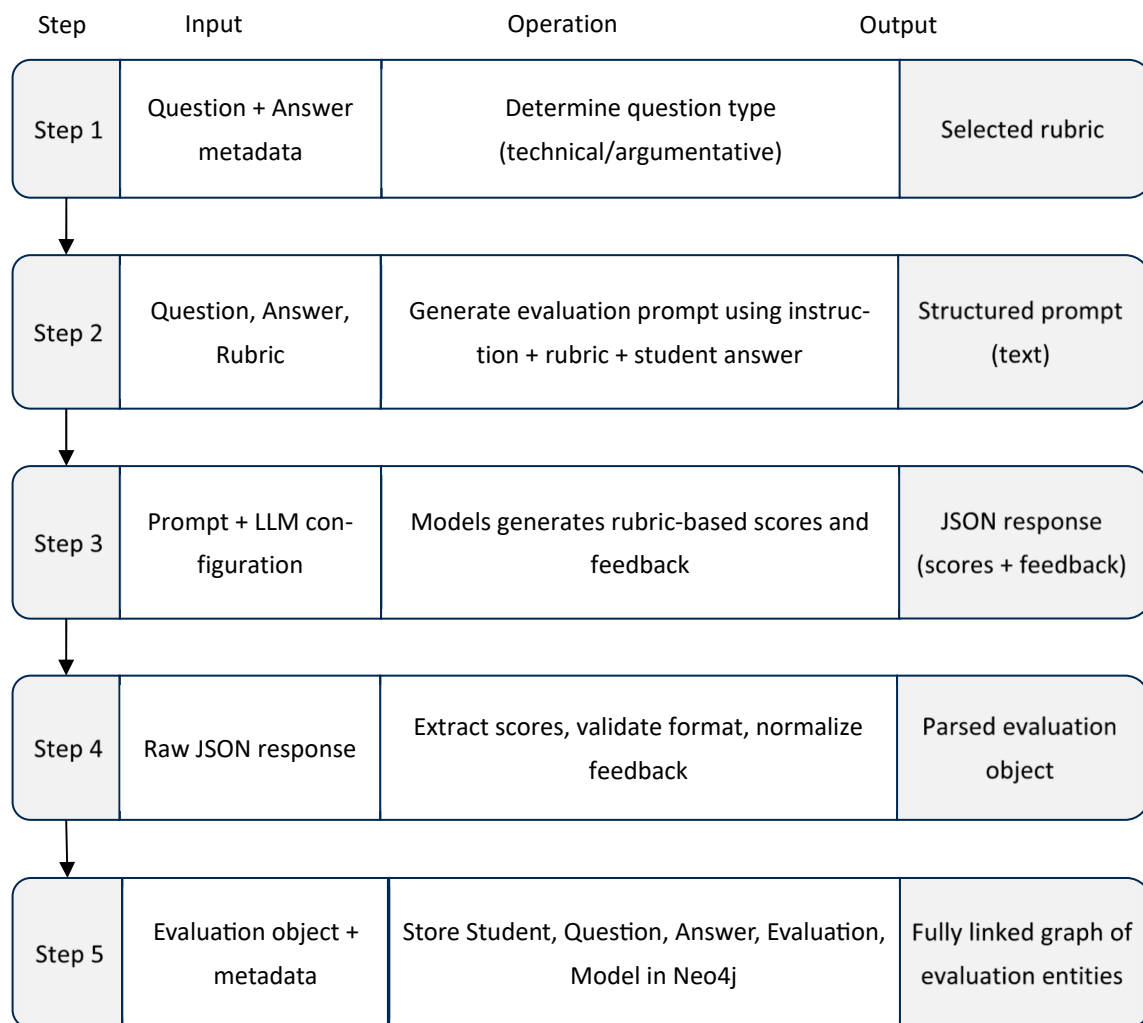


Figure 2. Detailed view of the five-stage evaluation pipeline, showing input, operation, and output at each step.

2.3. Model Configuration and Scoring Protocol

The evaluation framework supported multiple instruction-tuned large language models (LLMs), each configured independently through a modular interface. In this setting, open-source models acted as independent evaluators, each providing a unique judgment on the same student response. The proprietary GPT-4 model was accessed via API and designated as the expert evaluator, given its proven instruction-following reliability and strong alignment with human graders in rubric-based assessments [28]. All other models were evaluated in comparison to GPT-4, enabling analysis of score alignment, rubric adherence, and feedback consistency across evaluators.

Six open-source LLMs were deployed locally using the Ollama runtime, ensuring full control over inference parameters. The open-source set was selected to represent distinct model families and alignment styles while keeping parameter scale comparable (approximately 7–8B). The set included LLaMA3:8B, LLaMA3.1 [29], Nous-Hermes2 [30], OpenHermes [31], Dolphin-Mistral [32], and Gemma:7B-Instruct [33]. All models received the same structured prompt, which contained the rubric, question, and student answer, and were required to return output in a JSON-only format. To ensure transparency and reproducibility, the exact prompts used in all experiments—including rubric serialization and the JSON response schema—are available in the project’s GitHub repository (see Data Availability Statement).

To ensure fairness and consistency, all open-source models were configured with identical generation parameters. Specifically, the temperature was set to 0.0 to enforce deterministic output, and the context length was fixed at 8192 tokens to accommodate the full prompt content without truncation [34]. Inference was constrained to single-pass execution without self-correction loops, memory, or external access. Table 3 presents a comparative overview of the models used in this study, including their developer, approximate size, deployment method, generation parameters, and execution environment.

Table 3. Model characteristics, configuration parameters, and deployment details.

Model Name	Developer	Size	Deployment	Temperature	Context Length	Execution Environment
GPT-4	OpenAI	Unknown	API (OpenAI)	N/A	8192+	Cloud
LLaMA3:8B	Meta	8B	Ollama (local)	0.0	8192	Local
LLaMA3.1:latest	Meta	~7B	Ollama (local)	0.0	8192	Local
Nous-Hermes2	Nous	~7B	Ollama (local)	0.0	8192	Local
OpenHermes:latest	Teknium	~7B	Ollama (local)	0.0	8192	Local
Dolphin-Mistral:latest	Eric Hartford	~7B	Ollama (local)	0.0	8192	Local
Gemma:7B-Instruct	Google	~7B	Ollama (local)	0.0	8192	Local

To handle occasional formatting inconsistencies, all model outputs were automatically checked for JSON validity. In some cases, structural errors were detected, but all evaluation data were successfully recovered from the error logs. As a result, the dataset included a complete set of evaluations for all model–student–question combinations.

In addition to rubric-based outputs, the system recorded the response time for each model on every evaluation instance. This enabled runtime profiling and comparative analysis of inference efficiency across evaluators. All open-source models were executed on a dedicated CPU-based server equipped with dual Intel Xeon Gold 6252N processors at 2.30 GHz and 384 GB RAM, while GPT-4 evaluations were performed via OpenAI’s cloud infrastructure. All generated outputs, including scores, feedback, and associated metadata, were subsequently parsed, validated, and stored in a Neo4j graph database to ensure transparent, traceable, and reproducible evaluation [35].

2.4. Graph-Based Storage and Evaluation Traceability

To enable structured analysis, reproducibility, and traceability across all evaluation stages, the system stored all scoring data in a Neo4j graph database. Graph-based representations offered a natural structure for linking students, questions, answers, models, and evaluations [36].

Beyond storing scores and feedback, the graph schema represents evaluators as first-class nodes and records criterion-level tendencies as properties and relationships. This structure enables longitudinal tracking of each model’s evaluation profile and supports subsequent calibration against lecturer ratings while preserving full provenance.

We selected Neo4j rather than CSV and JSON and rather than relational databases because the evaluation corpus is intrinsically graph structured, with entities and dependencies most naturally modeled as nodes and edges, which enables path and provenance queries with stable referential integrity [37].

Rather than treating evaluations as isolated entries, the graph representation encoded the full context of each response. A student received a specific question and provided an open-ended answer. That answer was then evaluated by a large language model, which generated rubric-based scores and feedback. Each of these steps was stored as a distinct

node in the graph, connected by labeled relationships that reflected the flow of information through the system.

This design enabled fine-grained inspection of evaluation behavior: for instance, tracing how different models assessed the same student across questions, how rubric criteria were applied over time, or how feedback varied by evaluator. By embedding all entities and their interactions in a queryable graph, the system ensured auditability, supported exploratory analysis, and facilitated reproducible educational research [35,36].

Figure 3 illustrates the logical structure of the graph schema. Nodes represented the five core entities—Student, Question, Answer, Evaluation, and Model—while directed relationships captured how responses were generated, evaluated, and stored. Each evaluation was fully contextualized with respect to the original question, the student, and the model that produced it.

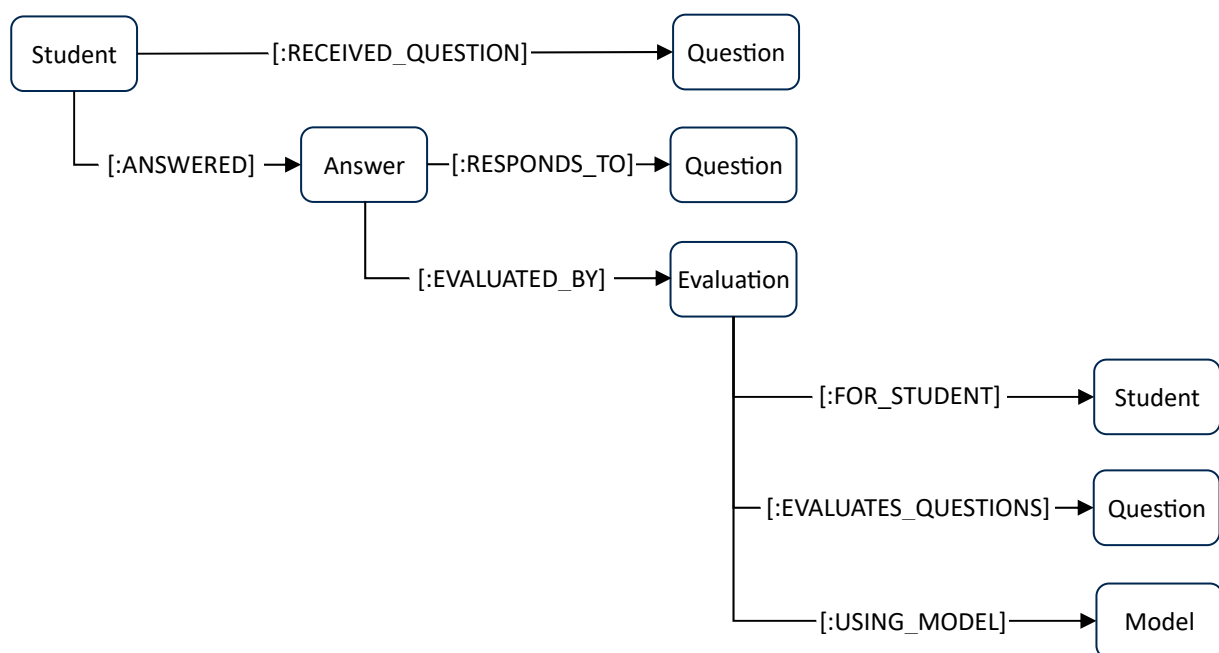


Figure 3. Logical structure of the Neo4j evaluation graph. The graph encodes the full assessment flow, linking each student, question, answer, evaluation, and model via semantically labeled relationships.

The methods outlined above defined a consistent and transparent pipeline for multi-model, rubric-guided evaluation of open-ended student responses. By combining structured prompt generation, controlled model execution, and graph-based storage, the system enabled reproducible assessment and criterion-level, cross-model analysis across students and tasks [38].

3. Results

Rubric-based scores were compared across models to identify systematic differences in scoring behavior for each evaluation criterion. Additional analyses included rubric coverage in feedback, semantic variation in evaluative language, inter-model agreement patterns, and execution consistency. Taken together, these results characterize the behavior of large language models acting as evaluators when constrained by uniform rubrics.

Rubric-based scores were compared across models to identify systematic differences in scoring at the criterion level. Additional analyses included rubric coverage in feedback, semantic variation in evaluative language, inter-model rank agreement, and execution consistency. Taken together, these results characterize the behavior of large language models acting as evaluators when constrained by uniform rubrics. To provide an immediate view of

evaluator generosity, Table 4 reports the overall mean score (1–10) for each model, averaged across all criteria and responses.

Table 4. Overall mean score by model across all criteria and responses (1–10), summarizing scoring generosity.

Model Name	Overall Mean Score
GPT-4	5.54
LLaMA3:8B	5.63
LLaMA3.1:latest	5.75
Nous-Hermes2	6.55
OpenHermes:latest	7.07
Dolphin-Mistral:latest	6.41
Gemma:7B-Instruct	6.72

3.1. Cross-Model Scoring Drift per Rubric Criterion

To examine whether different language models displayed consistent tendencies in their application of rubric criteria, the average scores assigned by each evaluator were computed for each criterion, separately for technical and argumentative tasks. Throughout this analysis, GPT-4 was treated as the expert reference, serving as an interpretive anchor for identifying overestimation or rubric misalignment in the outputs of other models. For clarity, models were displayed in descending order of overall average score in the heatmaps.

Across technical tasks, the most pronounced differences appeared in the accuracy and terminology dimensions. OpenHermes and Nous-Hermes2 produced consistently higher scores across all criteria, with mean values exceeding 7.5. Gemma and Dolphin-Mistral followed with slightly lower scores, while GPT-4 occupied one of the lowest positions in the ranking, assigning more conservative values overall—particularly for completeness (5.27) and accuracy (5.72). Models from the LLaMA3 family (8B and 3.1) reported moderate scores, slightly above GPT-4 on some criteria but consistently below the top-ranked models. These differences are illustrated in Figure 4, which presents the average scores per model and criterion for technical evaluations, ranked by total rubric score.

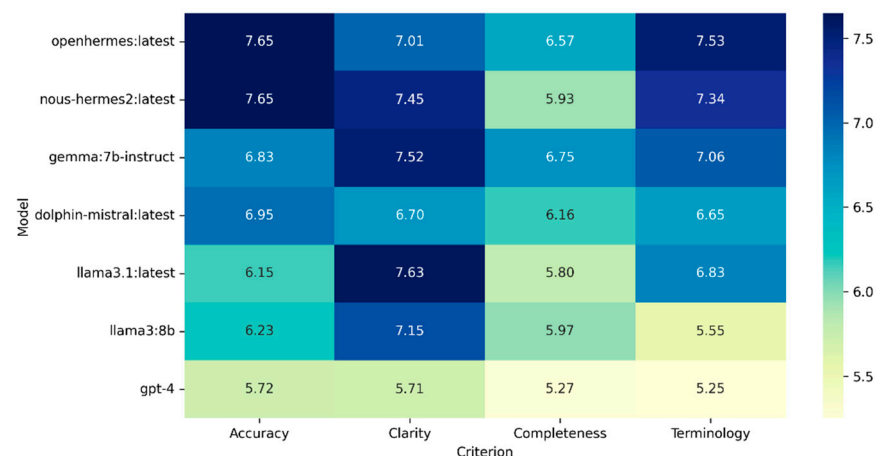


Figure 4. Average rubric scores per model for technical tasks.

For instance, GPT-4 assigned an average score of 5.72 for accuracy and 5.27 for completeness. These results demonstrated its conservative evaluation pattern. In contrast, Nous-Hermes2 yielded significantly higher averages, with 7.96 for accuracy and 7.78 for terminology. Dolphin-Mistral and Gemma also displayed relatively high scores across technical dimensions.

For argumentative responses, score divergence was even more pronounced, especially in the dialecticality and originality dimensions. OpenHermes again yielded the highest values across all criteria, including an average of 6.42 for originality. Gemma, Dolphin-Mistral, and Nous-Hermes2 followed with strong but slightly lower performance. GPT-4 appeared in the lower half of the ranking, with notably cautious scoring on coherence and dialecticality (4.81). This reflected a limited recognition of opposing viewpoints or counterarguments. The LLaMA3 models were positioned at the bottom, with average dialecticality scores under 3.2, indicating limited sensitivity to argumentative nuance in student responses. These comparative trends are visualized in Figure 5, which summarizes model-level averages for argumentative scoring dimensions.

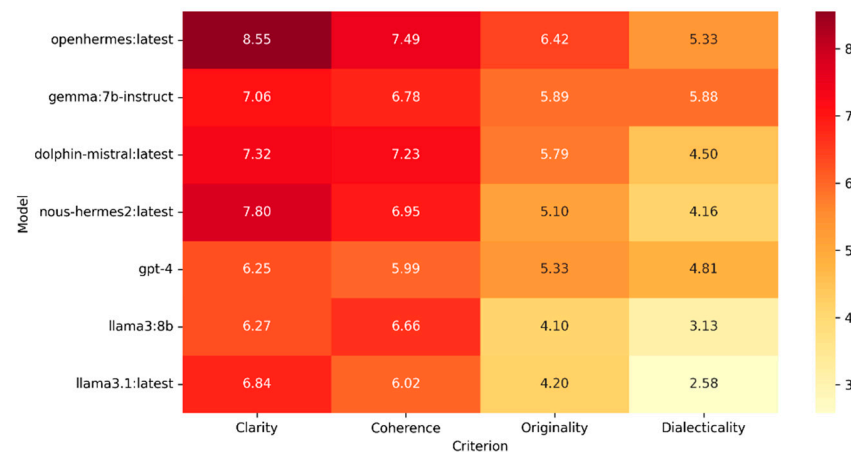


Figure 5. Average rubric scores per model for argumentative tasks.

Score divergence was especially visible for dialecticality. GPT-4 scored 4.81 on average, while Gemma reached 5.88 and Nous-Hermes2 reached 4.50. For originality, GPT-4 assigned 5.33, whereas Gemma and Dolphin-Mistral exceeded 5.8, suggesting that these models adopted a more generous interpretation of creative expression.

Although all models operated under identical rubric definitions and prompt structures, their scoring behavior reflected distinct judgment styles. When compared to the expert anchor (GPT-4), the observed scoring drifts indicated that some models systematically overestimated the quality of student responses. These differences reinforced the need for evaluator calibration and baseline alignment in multi-model automated assessment pipelines.

3.2. Rubric Alignment and Stylistic Variation in Model-Generated Feedback

To evaluate the extent to which rubric dimensions were explicitly reflected in the textual feedback generated by the models, we conducted a lexical analysis based on direct term matching. While numerical scoring provided structured assessment along predefined criteria, the clarity and transparency of feedback also depended on whether those criteria were linguistically referenced. By quantifying the presence of rubric terms in feedback, we estimated how explicitly each model communicated the evaluation structure in natural language form.

The dataset comprised 220 open-ended student responses, each evaluated independently by seven models (six open-source models and GPT-4), resulting in a total of 1540 feedback entries. For each feedback text, the content was converted to lowercase and scanned for exact matches with the rubric criterion names. The analysis was conducted separately for technical and argumentative questions. The technical rubric included the terms accuracy, clarity, completeness, and terminology, while the argumentative rubric consisted of clarity, coherence, originality, and dialecticality.

For each feedback entry, a binary presence value (0 or 1) was assigned to each criterion, depending on whether the corresponding term appeared at least once in the text. These values were then aggregated per model to compute two quantitative measures.

The first measure reflected the proportion of feedback responses that contained each rubric criterion name. These percentages were reported in Figure 6, which displays a matrix where each cell corresponds to a specific criterion–model pair.

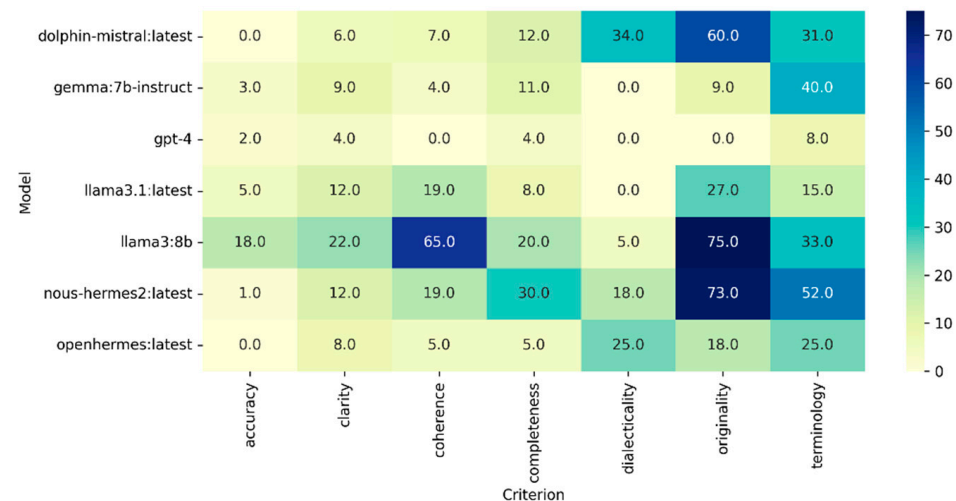


Figure 6. Proportion of feedback entries in which each rubric criterion name was explicitly mentioned, grouped by model. Criteria are grouped according to the corresponding rubric type (technical or argumentative).

The second measure represented the average number of rubric terms mentioned per feedback. It was calculated by summing the binary presence values for each feedback and averaging across all entries per model. The results were displayed in Figure 7, which illustrates comparative distributions across evaluators.

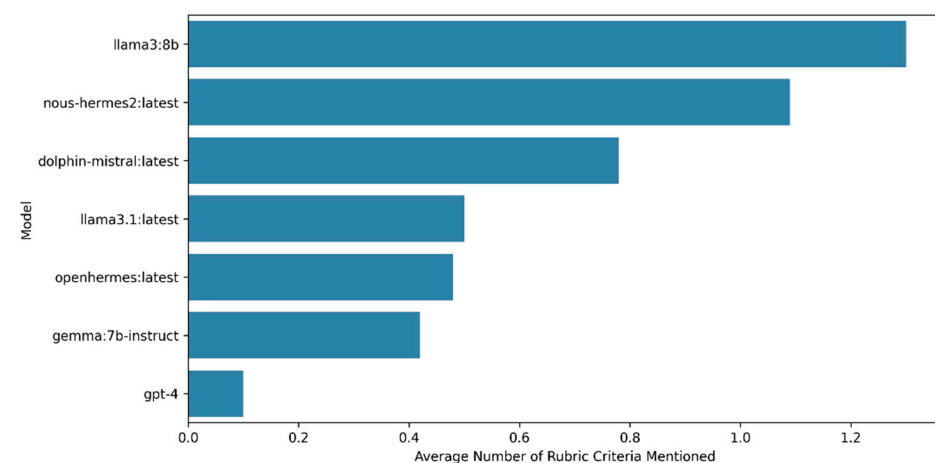


Figure 7. Average number of rubric criteria explicitly mentioned per feedback, aggregated by model across all evaluation instances.

While lexical analysis captured explicit mentions of rubric terms, it did not fully reflect the semantic content or evaluative depth of the feedback. The excerpts in Tables 4 and 5 illustrated these subtleties and served as a bridge toward the next layer of analysis—semantic divergence in model-generated feedback.

Table 5. Representative excerpts from seven LLMs evaluating the same technical student response.

Model	Feedback (Excerpt)
Gemma:7b-instruct	"The answer is somewhat vague. It doesn't clearly explain what a version control system is or how it helps manage changes over time."
LLaMA3.1:latest	"A version control system is not just about storing code, but also tracking changes, collaborating with others, and managing different versions."
Nous-Hermes2:latest	"The answer is partially accurate as it does not explain what a version control system actually does. It lacks clarity and completeness, as well as the use of proper terminology."
LLaMA3:8b	"This answer provides a good general understanding of version control systems but would benefit from more technical terminology and structured explanation."
OpenHermes:latest	"Your answer is mostly accurate but lacks specific examples and technical terminology. Including version history or branching would have improved completeness."
Dolphin-Mistral:latest	"This is a decent explanation, but it misses key aspects like change tracking and team collaboration. Completeness and terminology are limited."
GPT -4	"Your answer is partially accurate. It identifies the function of version control systems but lacks details on how changes are tracked and managed."

To further illustrate rubric alignment, Table 5 presents representative excerpts from feedback generated by seven different evaluators for the same technical student response. These fragments were selected from the full feedback texts to highlight variations in language, evaluative tone, and explicit reference to rubric dimensions. For instance, LLaMA3 offered a fluent explanation—"A version control system is not just about storing code. . ."—but did not reference any rubric dimension directly. In contrast, Nous-Hermes 2:latest evaluated rubric criteria explicitly, stating: "It lacks clarity and completeness, as well as the use of proper terminology." Meanwhile, OpenHermes:latest provided a balanced middle ground, identifying strengths ("mostly accurate") while suggesting improvements ("including version history. . ."). This diversity in style and rubric referencing illustrated how differently models interpreted and applied the same evaluation prompt, despite shared structure and criteria.

In parallel with the technical example, Table 6 presents representative excerpts from the feedback generated by seven evaluators for a single student response to an argumentative question. These fragments illustrated stylistic and structural differences in how each model interpreted rubric-aligned evaluation. Models such as Gemma:7b-instruct and LLaMA3:8b focused on surface-level clarity, whereas OpenHermes:latest and Nous-Hermes2:latest explicitly referred to missing counterarguments or dialectical complexity, which are key elements of the argumentative rubric. The excerpts were selected from longer feedback texts to emphasize rubric salience and variation in evaluative depth.

Table 6. Representative excerpts from seven LLMs evaluating the same argumentative student response.

Model	Feedback (Excerpt)
Gemma:7b-instruct	"The answer lacks a nuanced exploration of the topic. No counterpoints are considered, and the reasoning feels one-sided."
LLaMA3.1:latest	"Your answer is clear in its main points, but could benefit from more originality and consideration of alternative views."
Nous-Hermes2:latest	"The answer is clear and easy to understand. There is some attempt at structure, but no dialectical development or contrast of perspectives."

Table 6. Cont.

Model	Feedback (Excerpt)
LLaMA3:8b	“The answer is clear and easy to follow, but contains no critical discussion or originality. It reads more like a summary than an argument.”
OpenHermes:latest	“Your answer is clear and well-structured. However, it lacks depth and fails to address potential counterarguments.”
Dolphin-Mistral:latest	“The response makes a valid point but misses the opportunity to explore opposing arguments or demonstrate dialectical reasoning.”
GPT-4	“The answer is concise and grammatically correct but underdeveloped in terms of argumentative structure. It does not consider opposing views or justify claims.”

The lexical analysis—based on exact keyword matching between feedback and rubric terms—highlighted the frequency with which rubric dimensions were explicitly referenced in model-generated feedback. It also revealed consistent stylistic tendencies across evaluators. These patterns underscored the diversity in how LLMs formulated feedback when applying shared evaluation criteria.

3.3. Semantic Divergence in Feedback Across LLMs

To examine the semantic variation in the textual feedback produced by the models, we encoded each feedback entry as a dense vector representation (embedding) using Sentence-BERT [39]. The embeddings numerically captured the sentence-level meaning of each feedback message and allowed comparison of their semantic content. Each feedback embedding was a high-dimensional vector representing the underlying meaning of the model’s written feedback, which enabled similarity comparisons beyond surface-level lexical overlap. Semantic divergence was quantified using pairwise cosine distances between feedback embeddings, following established practices in semantic similarity analysis [40].

To explore the distribution of feedback content across evaluators, we applied t-distributed stochastic neighbor embedding (t-SNE) to reduce the high-dimensional Sentence-BERT embeddings to a two-dimensional plane [41]. The resulting axes, labeled *Component 1* and *Component 2*, are abstract coordinates generated by t-SNE to preserve local neighborhood structure from the embedding space. They do not correspond to rubric criteria or semantic dimensions, but provide a layout where relative distances indicate the degree of semantic similarity among feedback entries. Figure 8 illustrates two perspectives of this projection. When colored by model (Figure 8a), the points appear largely interspersed, showing that semantic content was broadly consistent across evaluators, with only low model-specific tendencies. In contrast, when colored by question (Figure 8b), two distinct patterns emerge. The technical questions (Q1–Q5) form compact and well-separated clusters, indicating stable and homogeneous feedback. The argumentative questions (Q6–Q10), however, are more dispersed and partly overlapping in the central region, reflecting greater semantic variability and higher similarity across prompts of this type. Overall, this analysis suggests that semantic variation in feedback is driven primarily by the nature of the question, being more tightly constrained for technical tasks and more heterogeneous for argumentative tasks.

To quantify inter-model semantic divergence at the task level, we computed cosine distances between feedback embeddings—numerical representations of the textual feedback generated by each model for the same student response. For each question, the feedback generated by all models was embedded using Sentence-BERT, which transformed each feedback text into a dense vector that captured its semantic content. All model pairs were then compared using cosine distance, a standard measure of dissimilarity in semantic space [42]. These distances were averaged across the dataset, resulting in a symmetric

matrix that reflected the typical semantic gap between each pair of evaluators. The lowest divergence occurred between Dolphin-Mistral and LLaMA3.1, with an average distance of less than 0.34, while the highest was observed between GPT-4 and LLaMA3:8B, exceeding 0.485, as shown in Figure 9.

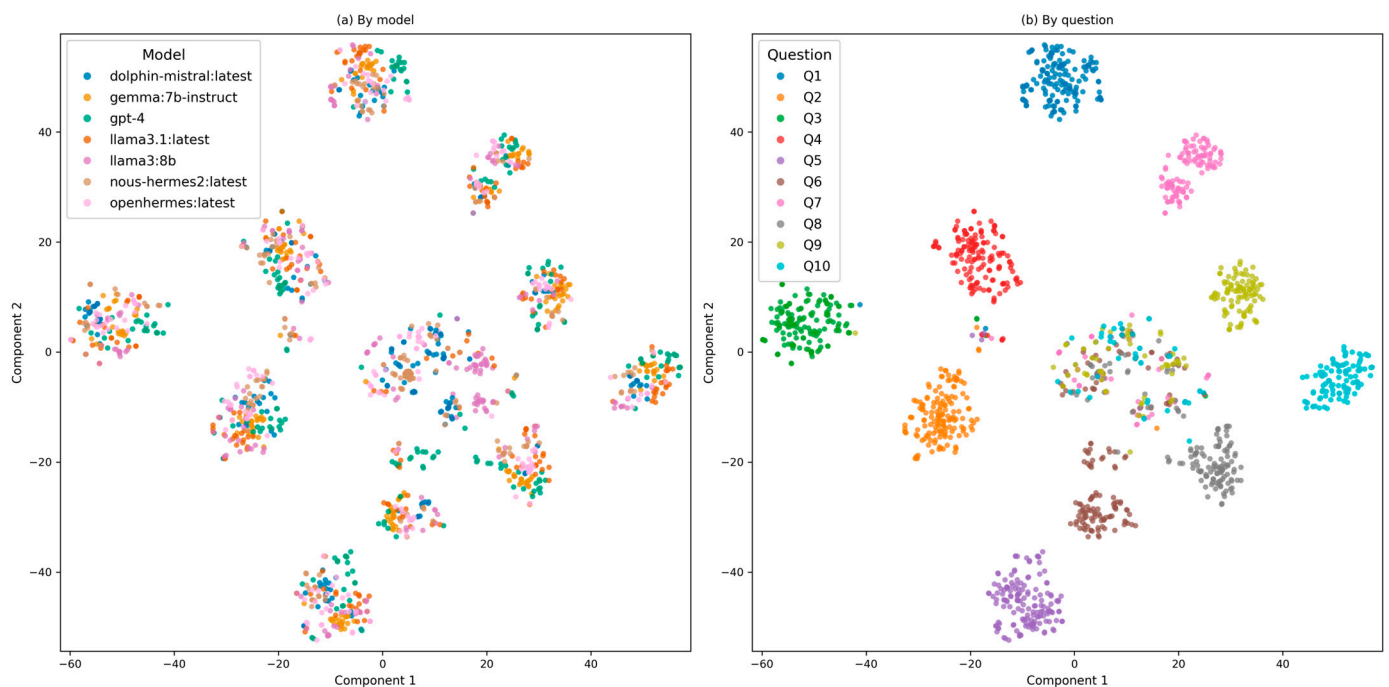


Figure 8. Two-dimensional t-SNE projection of Sentence-BERT embeddings of all feedback entries, visualized under two perspectives: (a) colored by model and (b) colored by question.



Figure 9. Average pairwise cosine distance between feedback embeddings across models. Lower values indicate greater semantic similarity.

GPT-4 generally exhibited greater divergence from open-source models, possibly reflecting differences in style, tone, or abstraction level. Taken together, these two analyses

provided evidence of meaningful variation in how different LLMs formulated feedback, even when evaluating the same answer under the same rubric. While overall feedback distributions substantially overlapped, the magnitude of pairwise semantic divergence highlighted differences in interpretive framing, specificity, and evaluative emphasis that may impact perceived clarity or fairness in educational settings.

3.4. Inter-Model Scoring Agreement Across Shared Rubric Criteria

To understand how different models interpret and apply the same evaluation rubric, we analyzed their scoring alignment on identical student responses. Since rubric-based scores are ordinal in nature, we used Spearman's rank correlation coefficient, a non-parametric statistic that measures the strength and direction of the monotonic relationship between two ranked variables, to quantify the level of agreement between pairs of models [43].

Given the differences in evaluative focus between the two rubrics—technical versus argumentative—we computed agreement separately for each rubric. For both cases, we first averaged the four rubric criteria per answer, per model, and then computed pairwise Spearman correlations across models. This allowed us to examine not only individual correlations but also the structural similarity of evaluative behavior across the model suite.

The correlation analysis revealed systematic alignment patterns across models. GPT-4 generally showed moderate agreement with the open-source models, reflecting its more conservative scoring tendencies. The highest correlations were observed among models from the same family (e.g., LLaMA3 variants), which displayed internally consistent scoring behavior. In contrast, models such as OpenHermes and Nous-Hermes2 diverged more strongly from GPT-4, indicating a tendency toward more generous evaluations. These findings suggest the presence of evaluator clusters within the model suite and highlight the importance of calibration when comparing or aggregating scores across heterogeneous LLM evaluators.

3.4.1. Technical Rubric

In technical evaluations, several models exhibited strong agreement in how they applied the rubric to student responses. The highest consistency was observed among Dolphin-Mistral:latest, OpenHermes:latest, and both LLaMA3 variants, with most pairwise correlation values exceeding 0.85. This suggested that these models shared similar evaluation tendencies when scoring factual accuracy, completeness, and terminology use. By contrast, GPT-4 showed weaker correlations with most open-source models, indicating a distinct, more conservative scoring profile. These patterns were reflected in the correlation matrix presented in Figure 10, which summarizes the pairwise rank correlations across models.

The inter-model alignment patterns were further clarified by the hierarchical clustering analysis of correlation values. As shown in Figure 11, Dolphin, OpenHermes, and the two LLaMA3 variants formed a compact cluster, indicating shared evaluation behavior. Meanwhile, GPT-4 and Gemma:7B-Instruct appeared more distant, reinforcing the distinctiveness of their judgment profiles.

In hierarchical clustering, larger linkage distances indicate lower concordance in the rank ordering of criterion-level scores across evaluators. GPT-4's separation is consistent with a more conservative technical scoring profile on accuracy and completeness under shared prompts and rubrics.

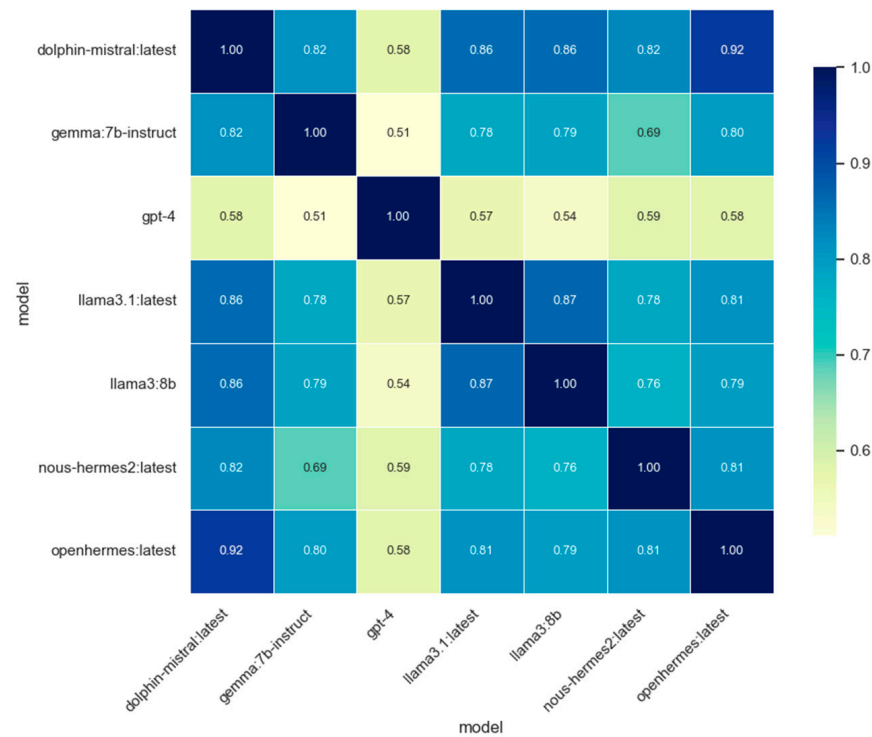


Figure 10. Spearman correlation matrix between models on the technical rubric.

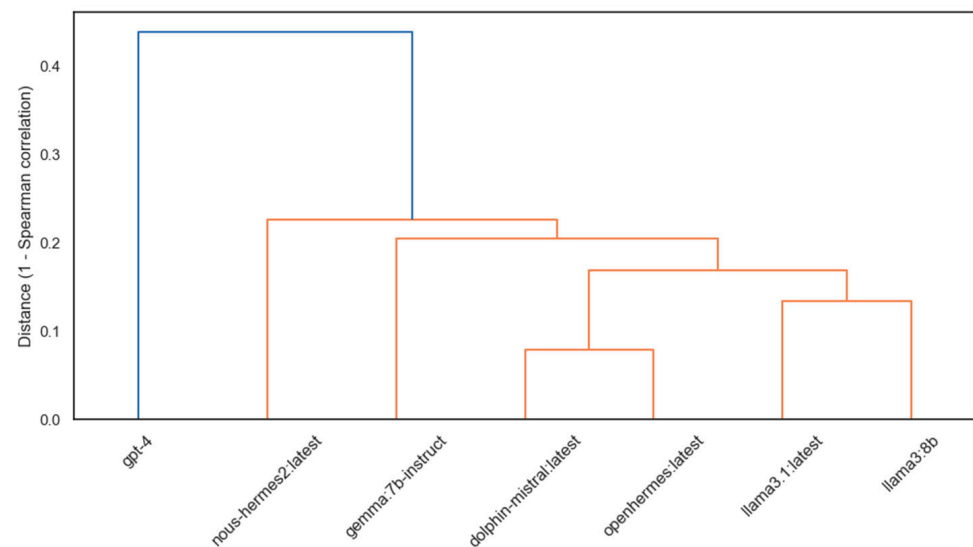


Figure 11. Hierarchical clustering diagram showing model similarity based on scoring behavior (technical rubric).

3.4.2. Argumentative Rubric

For argumentative tasks, overall agreement levels were lower. This outcome was expected, given the subjectivity of criteria such as coherence, originality, and dialecticality. Even so, certain alignment patterns persisted. Nous-Hermes2:latest and OpenHermes:latest maintained relatively strong correlations, while Gemma:7B-Instruct once again diverged from all other models. These correlation dynamics were detailed in Figure 12, which presents the corresponding matrix for the argumentative rubric.

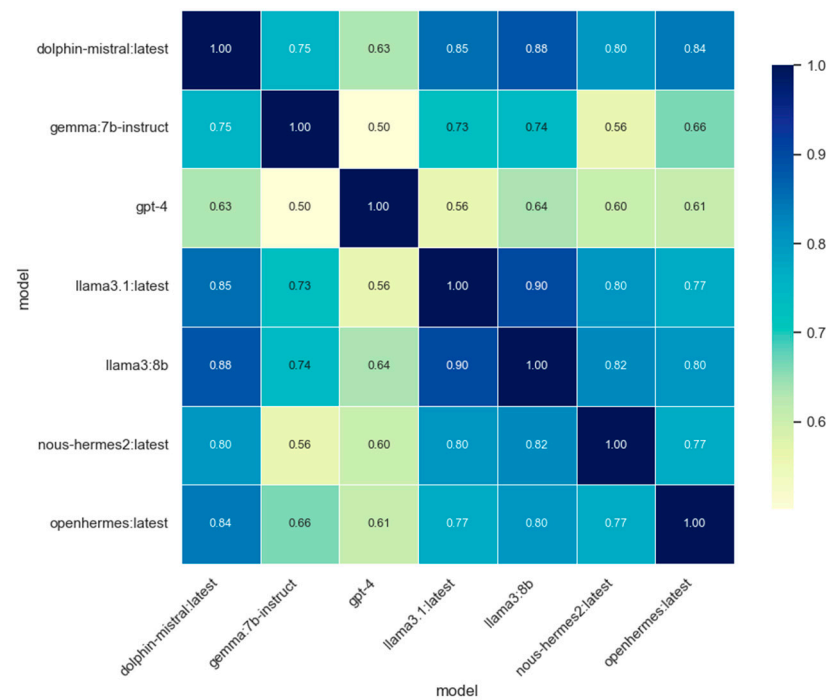


Figure 12. Spearman correlation matrix between models on the argumentative rubric.

The hierarchical clustering diagram in Figure 13 illustrates structural differences in model behavior when applying the argumentative rubric. GPT-4 clustered more closely with Dolphin and Nous-Hermes2, whereas the LLaMA3 models and Gemma appeared farther apart, indicating greater divergence in how evaluators interpreted open-ended, reasoning-intensive tasks.

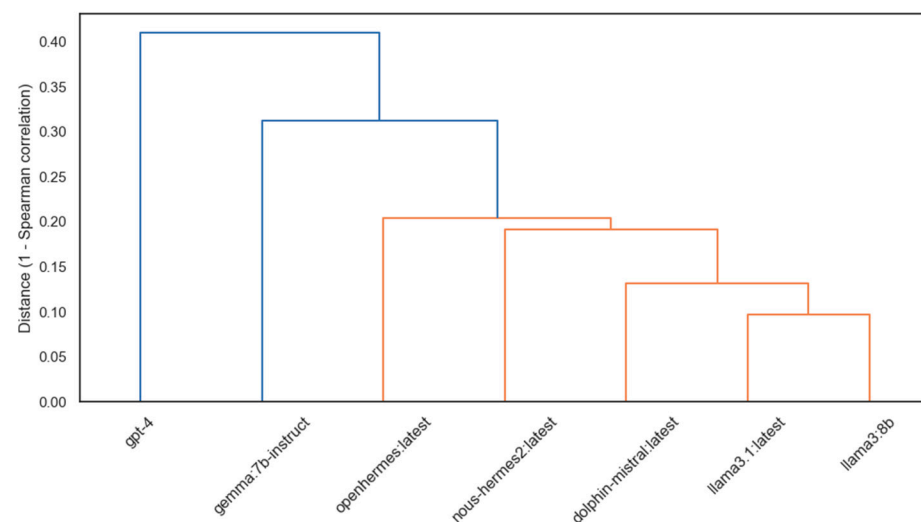


Figure 13. Hierarchical clustering diagram showing model similarity based on scoring behavior (argumentative rubric).

Conversely, shorter dendrogram distances denote greater similarity in relative ranking patterns, even when absolute score levels differ. Given the higher heterogeneity of argumentative criteria (e.g., originality, dialecticality), proximity in this rubric should be interpreted as convergence in ranking behaviour rather than equivalence of absolute scores.

Interpretation and Implications

The observed evaluator clusters suggested that some instruction-tuned models exhibited consistent judgment behavior across both rubrics, while others diverged significantly depending on task type. OpenHermes and Dolphin-Mistral consistently demonstrated high alignment, which may reflect similarities in training regimes or architectural design. Gemma, on the other hand, appeared as an outlier in both technical and argumentative contexts.

GPT-4, used here as the reference evaluator, showed only moderate agreement with the open-source models. This distinct behavior underlined the importance of evaluator selection and calibration when designing multi-model scoring pipelines. As emphasized in other critical domains such as medical diagnostics, failing to account for inter-evaluator variability may compromise the reliability and fairness of multi-model automated assessments [11–13,44].

3.5. Evaluation Latency and Response Length

To complement the rubric-based assessment analysis, we also examined two practical dimensions relevant to the deployment of automated evaluators: evaluation latency and response length (verbosity). These aspects were critical when considering the scalability, responsiveness, and interpretability of LLM-based grading systems in educational contexts. Measuring latency was particularly important when models were deployed in real-time or low-resource environments, where long inference times could reduce practical usability [45].

Figure 14 illustrates the average time required by each model to generate feedback and scores for a single student response. The fastest model was GPT-4, with an average evaluation time under 15 s, likely benefiting from optimized inference pipelines. Most other models, including Dolphin-Mistral:latest, LLaMA3:8B, and OpenHermes:latest, completed evaluations in approximately 25–30 s. The slowest models were Gemma:7B-Instruct and Nous-Hermes2:latest, each exceeding 35 s per response, likely due to larger output formats or slower inference pipelines.

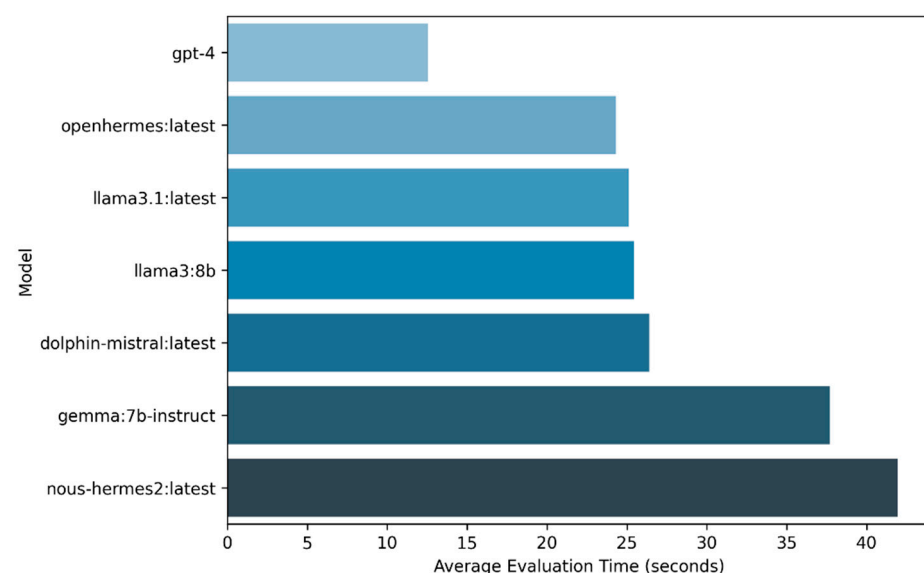


Figure 14. Average evaluation time per model (in seconds).

Figure 15 presents the average response length, measured by the word count of each model's generated feedback. All models evaluated 220 answers each. GPT-4 produced the most concise responses, averaging fewer than 30 words, while LLaMA3.1:latest and Gemma:7B-Instruct generated longer feedback responses, with averages close to 50 words.

The remaining models, such as Nous-Hermes2:latest and Dolphin-Mistral:latest, fell between these extremes, providing justifications of moderate length.

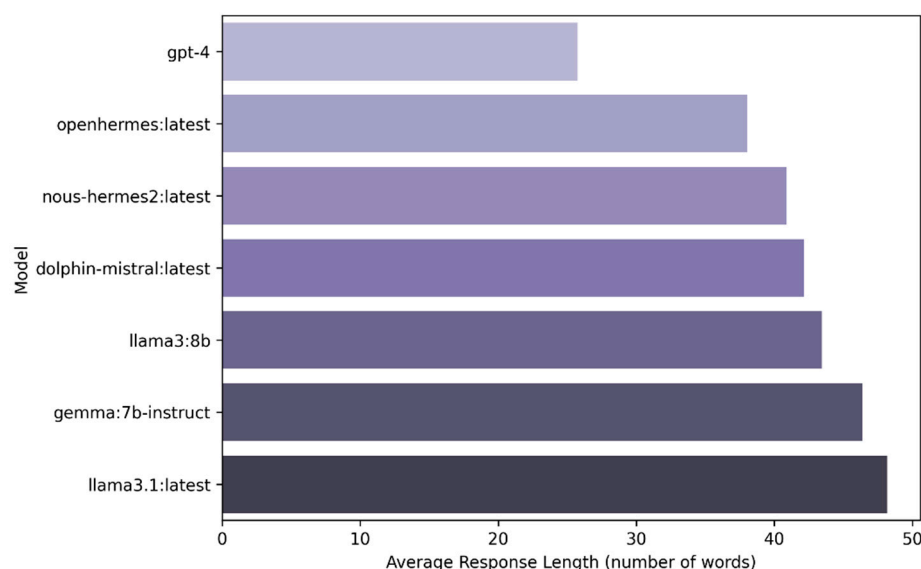


Figure 15. Average response length per model (number of words).

These latency and verbosity metrics offer complementary insights into the operational characteristics of each model. While shorter feedback and faster responses may be suitable for real-time or large-scale scenarios, longer outputs may enhance the interpretability and pedagogical usefulness of automated feedback. Thus, model selection may require balancing the trade-off between informativeness and efficiency, depending on the educational use case.

4. Discussion

The following discussion examines how large language models behaved when tasked with rubric-based evaluation and explored what the observed differences revealed about their consistency, evaluative style, and pedagogical suitability for automated educational assessment. These findings suggest that model choice and calibration are critical to ensuring fairness and interpretability in automated scoring systems.

4.1. Interpretation of Results

The rubric-based evaluations revealed consistent differences in how large language models applied scoring criteria, even under identical prompts and deterministic inference conditions. This variation was reflected both in the numerical scores assigned to student responses and in the qualitative feedback produced by each model.

In technical tasks, several models demonstrated a tendency toward higher overall scores, particularly OpenHermes:latest, Nous-Hermes2:latest, and Dolphin-Mistral:latest, which consistently rated answers above the reference model. In this context, GPT-4 was treated as an expert evaluator, serving as a high-quality baseline for interpreting score divergence and rubric fidelity. For instance, OpenHermes achieved average scores exceeding 8 in accuracy and clarity, while GPT-4 remained more conservative across all criteria, particularly on completeness (5.27) and terminology (5.44). These differences suggested that some models were more lenient in recognizing partial correctness or fluent expression, whereas others adopted a stricter interpretation of technical quality.

The contrast was even sharper in argumentative tasks. While OpenHermes again produced the most generous scores—leading on all four rubric criteria—GPT-4 assigned

lower values overall, particularly on coherence (5.99) and dialecticality (4.81). Gemma and Dolphin-Mistral showed higher appreciation for originality and stylistic nuance, while the LLaMA3 variants struggled to identify or reward dialectical engagement, with average scores often below 3. These results reinforced the observation that open-ended, subjective criteria introduced greater variability in scoring, making inter-model agreement more difficult to achieve in reasoning-intensive contexts.

Feedback analysis provided further evidence of model-specific evaluative styles. Some models, such as Nous-Hermes2, frequently referenced rubric terms explicitly, while others, including LLaMA3.1, offered more general suggestions with limited alignment to rubric language. GPT-4 generated the most semantically distinct feedback, as confirmed by embedding-based cosine distance analysis. Despite rarely using rubric terms directly, it offered concise and well-structured justifications, suggesting a more abstract and internally aligned interpretive process.

Despite prompt standardization and rubric enforcement, each model expressed a distinctive judgment pattern—visible both in scores and in feedback. These differences mattered: they implied that student responses could be evaluated and explained differently depending on the model used. Such variability challenged the reliability and fairness of multi-model systems and emphasized the need for careful evaluator calibration before deployment in educational contexts.

Results on evaluation latency and response length indicate a systematic trade-off between speed and conciseness versus informativeness and detail across evaluators. Under identical prompts and deterministic settings, GPT-4 produced shorter outputs at lower latency, supporting higher throughput, whereas several open-source models generated longer, more elaborate feedback at greater latency. These patterns delineate deployment choices: workflows that require rapid, large-scale scoring may favor concise evaluators, while formative contexts may benefit from models that prioritize explanatory breadth despite longer runtimes.

4.2. Implications for Automated Assessment

The results of this study have important implications for the use of large language models in educational assessment, particularly in open-ended, rubric-guided contexts. First, the consistent differences observed in scoring behavior and feedback formulation across models raise concerns about the reliability of LLM-based grading systems when used in isolation. Without evaluator calibration or external reference alignment, students may receive divergent scores or qualitatively different feedback for the same response, depending solely on which model is used. This undermines the principles of fairness and consistency that are fundamental to educational evaluation.

Second, the divergence was not limited to numerical scoring. Models varied in how explicitly they referred to rubric criteria, how they structured feedback, and what aspects of student reasoning they emphasized. Such differences may affect student perception, learning outcomes, and trust in automated systems. For example, concise but abstract feedback (as seen in GPT-4) may be misinterpreted or undervalued by students unfamiliar with implicit rubric reasoning, while verbose but rubric-referenced feedback (as in Nous-Hermes2 or OpenHermes) may offer clearer instructional guidance.

Third, the higher agreement observed among certain open-source models suggests that model family or training data similarity plays a role in scoring alignment. This opens the possibility of using model ensembles or voting schemes within a constrained family of evaluators to mitigate individual biases and improve robustness. Alternatively, system designers may opt to deploy a single calibrated model and prioritize transparency over diversity of judgment.

Ultimately, these findings highlight the importance of integrating model selection, rubric enforcement, and output normalization into the design of automated assessment tools. Large language models can serve as powerful evaluators, but their interpretive variability must be accounted for when used in real educational settings. This is particularly critical in formative assessment contexts, where feedback quality and score credibility directly influence learner development.

4.3. Methodological Considerations

The design of this study reflected a set of deliberate methodological choices. These choices were intended to ensure consistency, reproducibility, and fine-grained interpretability across models and responses. One such decision was the use of deterministic inference settings (temperature = 0.0) across all models. This eliminated variability across runs and allowed observed differences in output to be attributed solely to model behavior rather than stochastic sampling effects.

Another key design element was the use of predefined rubrics stored in structured JSON format. By clearly defining each evaluation criterion and using the same rubric for all models, the study ensured uniform task framing and enabled criterion-specific analysis of scores and feedback. The separation between technical and argumentative rubrics was essential for capturing the distinct demands of factual accuracy versus reasoning and rhetorical engagement.

All evaluations were parsed, normalized, and stored in a Neo4j graph database. This choice supported structured querying and traceability across entities such as student, question, answer, model, and evaluation. In contrast to flat CSV or JSON storage, the graph structure enabled longitudinal and criterion-level comparisons between evaluators and across responses, offering a more transparent and scalable approach to analysis.

Finally, the use of Sentence-BERT embeddings for semantic analysis provided a vector-based representation of feedback content, enabled quantitative comparison beyond surface-level text matching, and ensured relative comparability of semantic outputs through consistent application across models. Together, these methodological components formed an integrated and replicable pipeline for automated, multi-model evaluation of student writing.

4.4. Limitations

While this study aimed to create a consistent and transparent framework for rubric-based evaluation using large language models, several limitations should be acknowledged. First, no human graders were involved in scoring or feedback validation. As a result, the reliability of model outputs was measured only in relation to each other and to GPT-4 as a reference, rather than to expert-annotated ground truth. This limited the extent to which conclusions about scoring accuracy could be generalized. Accordingly, we do not claim absolute accuracy for any model and only characterise inter-model differences under shared rubrics.

Second, the evaluation relied on a fixed set of student responses to ten predefined questions, answered by a cohort of 22 undergraduate students. Although the questions were designed to cover both technical and argumentative domains, and the responses captured authentic variation in writing quality, the dataset remained relatively small in scope. This limited the statistical power of model comparisons and the generalizability of the findings to broader educational settings or other disciplines. Accordingly, cross-model contrasts are interpreted descriptively, and small effects may remain undetected.

Third, although prompts were carefully constructed to be clear and rubric-aligned, all models remained sensitive to prompt phrasing and input format. Small changes in wording or rubric structure could have led to different outputs, even under deterministic

inference settings. In addition, the feedback style and score generosity of each model were shaped by pretraining data and model-specific optimization, which were not accessible or controlled in this study.

Finally, the semantic similarity analysis using Sentence-BERT provided useful insight into feedback variation. However, embedding models such as Sentence-BERT, which were trained on general-purpose corpora, could have introduced interpretive bias. This might have led to an overemphasis on surface-level patterns or stylistic similarity rather than deeper evaluative meaning. As a result, they could not fully capture the educational value or pedagogical appropriateness of the feedback. Future research should consider triangulating embedding-based metrics with human judgments of helpfulness and alignment with rubric criteria.

Overall, this study should be regarded as a proof of concept for a reproducible, graph-based framework for multi-model educational evaluation. While the system demonstrated strong potential for fine-grained, rubric-aligned assessment, it had been tested only at a small scale and outside of real instructional workflows. Broader deployment and validation, including comparisons with expert human grading, remained necessary before operational adoption.

4.5. Future Work

Building on the current proof of concept, several directions can enhance both the scope and practical applicability of the proposed framework. One immediate extension will involve expanding the dataset to include a larger and more diverse pool of student responses, covering multiple disciplines and educational levels. This would enable broader generalization of model behavior and improve the robustness of inter-model comparisons. Additionally, incorporating human-graded samples would allow direct alignment with expert benchmarks, facilitating the calibration of model outputs to human expectations.

A key direction for future development will involve the creation of course-specific evaluators—language models that are fine-tuned on the content, terminology, and pedagogical structure of a particular academic course. Instead of applying generic instruction-tuned models, each evaluator would be tailored to a specific subject by integrating lecture materials, textbooks, instructor notes, and prior student assessments. This would enable models to better capture domain-specific expectations, technical vocabulary, and acceptable levels of detail, resulting in more accurate and pedagogically aligned evaluations.

In the long term, we envision a system in which each course is paired with a dedicated evaluator capable of scoring student responses and generating feedback in accordance with the instructor's style, curricular goals, and assessment philosophy. Such evaluators could support formative and summative assessment at scale, assist with grading consistency across sections, and provide targeted feedback to students in real time. Implementing this vision will require new methods for data curation, rubric adaptation, and lightweight fine-tuning, but it represents a promising step toward fully personalized, AI-supported educational evaluation.

5. Conclusions

This paper presented GraderAssist, a proof-of-concept framework for automated evaluation of open-ended responses using multiple large language models and rubric-based scoring. By integrating deterministic prompting, structured rubric definitions, and graph-based storage, the system enabled fine-grained, interpretable, and reproducible evaluation across heterogeneous models.

The results showed that language models differ not only in their scoring generosity but also in their alignment with rubric criteria and feedback formulation. While some models

exhibited strong internal consistency, others diverged substantially, particularly on subjective dimensions such as originality and dialecticality. Embedding-based semantic analysis confirmed that evaluator outputs varied in tone, abstraction, and rubric salience, even under identical prompts. These findings emphasize the importance of evaluator calibration and systematic benchmarking when deploying multi-model evaluation pipelines.

The main contribution lies in providing a replicable, multi-model evaluation pipeline that makes evaluator variability visible and measurable, supporting reproducibility and system calibration. Unlike prior work limited to single-model scoring, GraderAssist combines rubric-guided assessment, semantic feedback analysis, and graph-based traceability into a unified infrastructure. The inclusion of latency and verbosity profiling further demonstrates the scalability of the framework, allowing future studies to balance evaluation efficiency with feedback interpretability.

Beyond the educational case study, the framework highlights a generalizable design for transparent, reproducible, and scalable automated evaluation systems. Its modular structure and graph-based representation support applications in other domains where consistent, interpretable, and traceable AI-assisted judgments are required. In this sense, GraderAssist provides both a methodological foundation and a technical pathway for building next-generation evaluation pipelines that combine fairness, interpretability, and reproducibility at scale.

The graph-based backend also provides a practical substrate for course-specific evaluators: model selection and calibration can be conditioned on course, assignment, and criterion subgraphs, enabling targeted deployment with auditable traces.

Author Contributions: Conceptualization, C.A., A.A.A. and E.P.; methodology, C.A., M.V.C. and P.I.; software, A.S.B., C.A. and A.A.A.; validation, C.A., C.A.A., M.V.C., P.I. and E.P.; data curation, A.C. and A.S.B.; writing—original draft preparation, C.A., M.V.C. and P.I.; writing—review and editing, A.A.A., E.P. and A.C.; visualization, A.S.B. and A.C.; supervision, C.A., M.V.C., P.I. and C.A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The source code of the main modules and the anonymized dataset are available at: https://github.com/anghelcata/auto_grader.git (accessed on 17 September 2025).

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

References

1. Mistler, W. Formal Assessment in STEM Higher Education. *J. Tech. Stud.* **2025**, *1*, 429. [CrossRef]
2. French, S.; Dickerson, A.; Mulder, R.A. A review of the benefits and drawbacks of high-stakes final examinations in higher education. *High. Educ.* **2024**, *8*, 893–918. [CrossRef]
3. Cheng, Y.; Li, X.; Wang, Q.; Zhang, W. LUPDA: A comprehensive rubrics-based assessment model. *Int. J. STEM Educ.* **2025**, *12*, 21. [CrossRef]
4. Tan, L.Y.; Hu, S.; Yeo, D.J.; Cheong, K.H. A Comprehensive Review on Automated Grading Systems in STEM Using AI Techniques. *Mathematics* **2025**, *13*, 2828. [CrossRef]
5. Souza, M.; Margalho, É.; Lima, R.M.; Mesquita, D.; Costa, M.J. Rubric's Development Process for Assessment of Project Management Competences. *Educ. Sci.* **2022**, *12*, 902. [CrossRef]
6. Emirtekin, E. Large Language Model-Powered Automated Assessment: A Systematic Review. *Appl. Sci.* **2025**, *15*, 5683. [CrossRef]
7. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training language models to follow instructions with human feedback. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS 2022), New Orleans, LA, USA, 28 November–9 December 2022. Available online: https://proceedings.neurips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html (accessed on 17 September 2025).
8. Mendonça, P.C.; Quintal, F.; Mendonça, F. Evaluating LLMs for Automated Scoring in Formative Assessments. *Appl. Sci.* **2025**, *15*, 2787. [CrossRef]

9. Spearman, C. The Proof and Measurement of Association between Two Things. *Am. J. Psychol.* **1904**, *15*, 72–101. [CrossRef]
10. Wei, Y.; Zhang, R.; Zhang, J.; Qi, D.; Cui, W. Research on Intelligent Grading of Physics Problems Based on Large Language Models. *Educ. Sci.* **2025**, *15*, 116. [CrossRef]
11. Iacobescu, P.; Marina, V.; Anghel, C.; Anghele, A.-D. Evaluating Binary Classifiers for Cardiovascular Disease Prediction: Enhancing Early Diagnostic Capabilities. *J. Cardiovasc. Dev. Dis.* **2024**, *11*, 396. [CrossRef]
12. Anghele, A.-D.; Marina, V.; Dragomir, L.; Moscu, C.A.; Anghele, M.; Anghel, C. Predicting Deep Venous Thrombosis Using Artificial Intelligence: A Clinical Data Approach. *Bioengineering* **2024**, *11*, 1067. [CrossRef] [PubMed]
13. Dragosloveanu, S.; Vulpe, D.E.; Andrei, C.A.; Nedelea, D.-G.; Garofil, N.D.; Anghel, C.; Dragosloveanu, C.D.M.; Cergan, R.; Scheau, C. Predicting periprosthetic joint Infection: Evaluating supervised machine learning models for clinical application. *J. Orthop. Transl.* **2025**, *54*, 51–64. [CrossRef]
14. Cisneros-González, J.; Gordo-Herrera, N.; Barcia-Santos, I.; Sánchez-Soriano, J. JorGPT: Instructor-Aided Grading of Programming Assignments with Large Language Models (LLMs). *Future Internet* **2025**, *17*, 265. [CrossRef]
15. Pan, Y.; Nehm, R.H. Large Language Model and Traditional Machine Learning Scoring of Evolutionary Explanations: Benefits and Drawbacks. *Educ. Sci.* **2025**, *15*, 676. [CrossRef]
16. Seo, H.; Hwang, T.; Jung, J.; Namgoong, H.; Lee, J.; Jung, S. Large Language Models as Evaluators in Education: Verification of Feedback Consistency and Accuracy. *Appl. Sci.* **2025**, *15*, 671. [CrossRef]
17. Monteiro, J.; Sá, F.; Bernardino, J. Experimental Evaluation of Graph Databases: JanusGraph, Nebula Graph, Neo4j, and TigerGraph. *Appl. Sci.* **2023**, *13*, 5770. [CrossRef]
18. Panadero, E.; Jonsson, A. The use of scoring rubrics for formative assessment purposes revisited: A review. *Educ. Res. Rev.* **2013**, *9*, 129–144. [CrossRef]
19. Chan, C.-M.; Chen, W.; Su, Y.; Yu, J.; Xue, W.; Zhang, S.; Fu, J.; Liu, Z. ChatEval: Towards Better LLM-based Evaluators through Multi-Agent Debate. In Proceedings of the Twelfth International Conference on Learning Representations (ICLR), Vienna, Austria, 7–11 May 2024. Available online: <https://openreview.net/forum?id=FQepisCUWu> (accessed on 17 September 2025).
20. Li, T.; Chiang, W.-L.; Frick, E.; Dunlap, L.; Zhu, B.; Gonzalez, J.E.; Stoica, I. From Live Data to High-Quality Benchmarks: The Arena-Hard Pipeline. Available online: <https://lmsys.org/blog/2024-04-19-arena-hard/> (accessed on 12 September 2025).
21. Bhat, V. RubricEval: A Scalable Human-LLM Evaluation Framework for Open-Ended Tasks. Available online: <https://web.stanford.edu/class/cs224n/final-reports/256846781.pdf> (accessed on 31 October 2025).
22. Hashemi, H.; Eisner, J.; Rosset, C.; Van Durme, B.; Kedzie, C. LLM-RUBRIC: A Multidimensional, Calibrated Approach to Automated Evaluation of Natural Language Texts. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL), Bangkok, Thailand, 11–16 August 2024; pp. 13806–13834. [CrossRef]
23. Feng, Z.; Zhang, Y.; Li, H.; Wu, B.; Liao, J.; Liu, W.; Lang, J.; Feng, Y.; Wu, J.; Liu, Z. TEaR: Improving LLM-based Machine Translation with Systematic Self-Refinement. In Proceedings of the Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, NM, USA, 29 April–4 May 2025; pp. 3922–3938. [CrossRef]
24. Shinn, N.; Cassano, F.; Gopinath, A.; Narasimhan, K.; Yao, S. Reflexion: Language Agents with Verbal Reinforcement Learning. In Proceedings of the Advances in Neural Information Processing Systems 36—NeurIPS, New Orleans, LA, USA, 28 November–9 December 2023.
25. Zhao, R.; Zhang, W.; Chia, Y.K.; Zhao, D.; Bing, L. Auto Arena of LLMs: Automating LLM Evaluations with Agent Peer-Battles and Committee Discussions. Available online: <https://auto-arena.github.io/blog/> (accessed on 12 September 2025).
26. Cantera, M.A.; Arevalo, M.-J.; García-Marina, V.; Alves-Castro, M. A Rubric to Assess and Improve Technical Writing in Undergraduate Engineering Courses. *Educ. Sci.* **2021**, *11*, 146. [CrossRef]
27. Shahriar, S.; Lund, B.D.; Mannuru, N.R.; Arshad, M.A.; Hayawi, K.; Bevara, R.V.K.; Mannuru, A.; Batool, L. Putting GPT-4o to the Sword: A Comprehensive Evaluation of Language, Vision, Speech, and Multimodal Proficiency. *Appl. Sci.* **2024**, *14*, 7782. [CrossRef]
28. Zhang, D.-W.; Boey, M.; Tan, Y.Y.; Jia, A.H.S. Evaluating large language models for criterion-based grading from agreement to consistency. *npj Sci. Learn.* **2024**, *9*, 79. [CrossRef]
29. Meta. Introducing Meta Llama 3. Available online: <https://ai.meta.com/blog/meta-llama-3/> (accessed on 17 September 2025).
30. Nous Research. Nous-Hermes 2 Model Card. Available online: <https://ollama.com/library/nous-hermes2:latest> (accessed on 17 September 2025).
31. Teknum. OpenHermes Model Card. Available online: <https://ollama.com/library/openhermes> (accessed on 17 September 2025).
32. Hartford, E. Dolphin-Mistral Model Card. Available online: <https://ollama.com/library/dolphin-mistral> (accessed on 17 September 2025).
33. Google. Gemma:7B-Instruct Model Card. Available online: <https://ollama.com/library/gemma:7b-instruct> (accessed on 17 September 2025).
34. Renze, M. The Effect of Sampling Temperature on Problem Solving in Large Language Models. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, FL, USA, 12–16 November 2024; pp. 7346–7356. [CrossRef]

35. Dong, B.; Bai, J.; Xu, T.; Zhou, Y. Large Language Models in Education: A Systematic Review. In Proceedings of the 2024 6th International Conference on Computer Science and Technologies in Education (CSTE), Xi'an, China, 19–21 April 2024. [CrossRef]
36. Mazein, I.; Rougny, A.; Mazein, A.; Henkel, R.; Gütebier, L.; Michaelis, L.; Ostaszewski, M.; Schneider, R.; Satagopam, V.; Jensen, L.J.; et al. Graph databases in systems biology: A systematic review. *Brief. Bioinform.* **2024**, *25*, bbae561. [CrossRef] [PubMed]
37. Asplund, E.; Sandell, J. Comparison of Graph Databases and Relational Databases Performance. Available online: <https://su.diva-portal.org/smash/record.jsf?pid=diva2:1784349> (accessed on 31 October 2025).
38. Chen, M.; Poulsen, S.; Alkhabaz, R.; Alawini, A. A Quantitative Analysis of Student Solutions to Graph Database Problems. In Proceedings of the the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1, Virtual Event, Germany, 26 June–1 July 2021; pp. 283–289. [CrossRef]
39. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 3982–3992. [CrossRef]
40. Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K.Q.; Artzi, Y. BERTScore: Evaluating Text Generation with BERT. In Proceedings of the International Conference on Learning Representations (ICLR 2020), Addis Ababa, Ethiopia (Virtual), 26 April–1 May 2020; Available online: <https://openreview.net/forum?id=SkeHuCVFDr> (accessed on 17 September 2025).
41. Poličar, P.G.; Stražar, M.; Zupan, B. openTSNE: A Modular Python Library for t-SNE Dimensionality Reduction and Embedding. *J. Stat. Softw.* **2024**, *109*, 1–30. [CrossRef]
42. Gómez, J.; Vázquez, P.-P. An Empirical Evaluation of Document Embeddings and Similarity Metrics for Scientific Articles. *Appl. Sci.* **2022**, *12*, 5664. [CrossRef]
43. Gauthier, T.D. Detecting Trends Using Spearman's Rank Correlation Coefficient. *Environ. Forensics* **2001**, *2*, 359–362. [CrossRef]
44. Vulpe, D.E.; Anghel, C.; Scheau, C.; Dragosloveanu, S.; Săndulescu, O. Artificial Intelligence and Its Role in Predicting Periprosthetic Joint Infections. *Biomedicines* **2025**, *13*, 1855. [CrossRef]
45. Lazuka, M.; Anghel, A.; Parnell, T.P. LLM-Pilot: Characterize and Optimize Performance of your LLM Inference Services. In Proceedings of the SC '24: International Conference for High Performance Computing, Networking, Storage and Analysis, Atlanta, GA, USA, 17–22 November 2024. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.