

Article

LLMs in Automated Assessment: A Role-Based Taxonomy and Framework for Controlled Educational Integration

Anastasia Vangelova  and Veska Gancheva * 

Department of Programming and Computer Technologies, Faculty of Computer Systems and Technologies, Technical University of Sofia, 1756 Sofia, Bulgaria; avangelova@yahoo.com

* Correspondence: vgan@tu-sofia.bg

Abstract

Large language models (LLMs) are reshaping the automated assessment of open-ended student responses. Compared with earlier rule-based, statistical, and feature-engineered approaches, they enable a deeper interpretation of meaning, context, and argumentation. This development can be understood as a fifth generation of automated scoring systems, but it also raises a new question: not only what LLMs can do, but also how they can be deployed in education in a controlled and reliable manner. This paper presents a role-based taxonomy that distinguishes between generative LLMs used as direct virtual graders, encoder transformers used as semantic tools, and intermediate text-to-text models used in more formalized assessment tasks. It also discusses the main limitations of standalone LLM graders, including hallucinations, probabilistic instability, limited interpretability, bias, and weak grounding in domain-specific content. To address these issues, the paper presents a developed framework implemented in an integrated assessment system built on role prompting, rubric-constrained grading, Retrieval-Augmented Generation (RAG), structured machine-readable outputs, workflow orchestration, and LMS integration. The framework is further extended to multimodal assessment through vision-based evaluation of visual artifacts such as UML state diagrams. The main contribution of the paper is not only a conceptual framework, but also its realization in a working integrated system for automated assessment in a more traceable, pedagogically grounded, and institutionally reliable way.

Keywords: automated assessment; large language models; open-ended questions; RAG; semantic analysis



Academic Editors: Jose Maria Alvarez Rodriguez and Douglas O'Shaughnessy

Received: 7 April 2026

Revised: 14 June 2026

Accepted: 30 June 2026

Published: 2 July 2026

Copyright: © 2026 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Automated assessment of open-ended questions is increasingly important at the intersection of educational technology and artificial intelligence. Unlike closed-ended test tasks, where the assessment can be based on a single correct answer, open-ended questions require interpretation of meaning, reasoning, logical consistency, conceptual depth, and the degree of coverage of predefined criteria. This interpretive complexity makes them pedagogically valuable, but also significantly more difficult to scale and to assess consistently and transparently [1,2]. In a university context, this problem also has a clear organizational dimension. Manual assessment of free-text answers is associated with significant time constraints, variability between assessors, and limitations in the timeliness and granularity of feedback, especially in mass courses and in more frequent assessment

activities [3,4]. Therefore, automation is not just a technical convenience, but a response to a systemic problem related to the scalability, consistency, and sustainability of assessment.

The development of automated scoring shows a gradual transition from symbolic, rule-based, and feature-based approaches to statistical methods, classical machine learning models, and subsequently to deep neural architectures. Earlier generations made important contributions to the formalization of the scoring process, but remained limited either by their dependence on manually defined features or by their difficulties in capturing more complex semantic and argumentative dependencies. Deep learning models significantly improve the work with free text by automatically extracting representations from raw content, but do not completely overcome the problems of interpretability, portability, and dependence on large annotated corpora [5]. In this study, these previous stages are considered a necessary background that prepares the methodological and technological foundation for the modern development of the field.

The current stage can be defined as the fifth generation of automated scoring systems, dominated by transformer-based architectures and large language models (LLMs). Unlike previous approaches based on predefined features or sequential neural representations, LLMs enable broader contextual, semantic and argumentative interpretation of student responses [6,7]. Recent research has shown that models such as GPT, Claude, Gemini, and various open LLMs can achieve levels of agreement with human raters comparable to inter-rater consistency between two experts, especially when clear analytical rubrics and structured prompting strategies are used [3,8]. This creates a genuine opportunity for LLMs to be used not only as aids, but also as a core intelligent layer in automated scoring. However, along with this potential, significant risks also emerge: hallucinations, probabilistic instability, sensitivity to the wording of the prompt, limited interpretability, risk of bias, and uncertain fidelity of the generated explanations [8,9].

Thus, the main challenge in the fifth generation of automated scoring systems can be formulated as follows: the increase in semantic power is accompanied by a decrease in control over the assessment process. The more capable the model is of interpreting complex text, the more acute the questions of validity, reproducibility, traceability, and pedagogical validity of the assigned score arise [6,10,11]. This is especially problematic when the LLM functions as a stand-alone grader without contextual grounding, without a clearly defined rubric framework, and without a machine-verifiable output structure [12–14].

In response to these limitations, research has gradually shifted to architectural control strategies that constrain and structure the role of the LLM in assessment. Among the most significant of these are rubric-constrained prompting, in which the model assigns grades within an analytical rubric; Retrieval-Augmented Generation (RAG), which grounds the analysis in specific learning materials; structured outputs in formats such as JSON, which make the output machine-verifiable and traceable; and workflow orchestration and LMS integration, through which assessment is embedded in a real institutional process. An additional line of development is represented by instrumentalized agents and hybrid architectures, in which the language model is supported by external tools for logical, arithmetic, or structural verification [15,16]. Despite the growing number of publications, the literature remains fragmented. Some studies compare models, others examine the effect of specific prompting strategies, and still others focus on RAG or on individual aspects of automatically generated feedback. Less frequently, a comprehensive conceptual framework is proposed that simultaneously encompasses the roles of models, the main risks of the fifth generation, and architectural strategies for their reliable implementation in an educational context. Even more rarely, these elements are associated with real LMS environments, where issues of process integration, output formalization, and institutional traceability are critical.

Despite the rapidly expanding body of research on LLM-assisted grading, current studies remain predominantly focused on isolated performance benchmarking, prompt engineering effectiveness, or general discussions of pedagogical opportunity. Comparatively little attention has been devoted to the operational question of how different functional roles assumed by LLMs in assessment generate distinct reliability risks, and how these risks can be mitigated through a controlled educational integration architecture. As a result, the field still lacks a unified framework connecting role differentiation, reliability governance, and institutionally deployable implementation.

The present study addresses this gap and focuses first on the role-based taxonomy and on the architectural principles for controlled educational integration of LLMs in assessment. Second, the paper proposes a controlled framework for educational integration that constrains grading-related model autonomy through six interrelated components: role prompting, rubric-constrained grading, RAG-based contextual grounding, structured machine-readable outputs, workflow orchestration, and LMS integration.

On this basis, the paper also provides a role-based systematization of contemporary model use in assessment workflows and examines how these architectural components can be combined into a more traceable, pedagogically grounded, and institutionally reliable assessment process.

The implemented scoring layer underlying this framework has already been empirically validated in complementary work in a real university setting, where AI-generated scores were compared with independent human scoring using agreement, reliability, correlation, and error metrics [17]. In addition, preliminary expert-based evaluation of the pedagogical quality of the generated feedback, reported in complementary work, showed high ratings for clarity, usefulness, and accuracy. These results provide empirical support for the practical viability of the implemented system, while the present article concentrates on the conceptual, architectural, and integration-oriented contribution.

In this sense, the contribution of the paper does not lie in proposing a new foundation model, but in formulating and operationalizing a controlled educational architecture for the use of LLMs in automated assessment, that connects functional taxonomy, reliability-oriented design principles, and real deployment logic for fifth-generation automated assessment.

Accordingly, the investigation is guided by the following research questions:

RQ1: How can the functional roles of LLMs in automated assessment be systematically classified from an educational deployment perspective?

RQ2: What architectural control mechanisms are required to ensure reliable institutional integration of these roles into assessment workflows?

RQ3: To what extent does the implemented controlled workflow demonstrate practical viability through agreement with expert grading, structured traceability, and exploratory multimodal assessment in a real LMS-connected educational setting?

2. Related Work

Automated open-ended response scoring has evolved in stages, from systems based on explicit rules to models capable of dealing with semantics, context, and argumentation. Earlier generations are important because they provide a methodological foundation for understanding contemporary approaches.

The first generation includes symbolic, rule-based, and ontological models, in which scoring is derived from predefined features and rules, as in the early tradition represented by PEG and related foundational work [18,19]. The second generation moves to statistical and feature-based approaches that model the relationship between linguistic indicators and human ratings and later support early operational systems such as e-rater [20–22].

The third generation introduces supervised machine learning, in which the score is predicted from manually constructed features using algorithms such as SVM, Random Forest, and boosting models [5,23]. The fourth generation is related to deep learning and automatic extraction of representations from raw text using CNN, LSTM, BiLSTM, and attention-based architectures [5].

The fifth generation of automated scoring systems is characterized by the dominant presence of transformer-based architectures and large language models (LLMs), which process text through the self-attention mechanism and build a global contextual representation of the input [7]. Unlike previous generations, in which scoring was based on predefined features or on sequential neural representations, modern models can perform more complex semantic, logical, and argumentative interpretations.

In the context of automated scoring, this development requires that models be considered not only according to their internal architecture, but also according to the function they perform in the scoring process itself. In this study, contemporary models are grouped according to their primary operational role in assessment workflows rather than by a rigid one-to-one correspondence between architecture and function. This distinction is important because the same model family may occupy different roles depending on how it is embedded in the assessment pipeline. For example, generative models may function as direct virtual graders when they produce scores, justifications, and feedback, but they may also act as semantic tools when used only for embeddings, retrieval, or auxiliary processing. Similarly, encoder-based transformers typically function as semantic similarity and representation tools, yet when fine-tuned to predict scores directly they can operate as graders in practical assessment settings. Table 1 summarizes this role-based taxonomy and highlights both the dominant functions and the main boundary cases of the different model families.

Table 1. Taxonomy of contemporary models for automated assessment according to architecture and primary operational role in the assessment process.

Model Family	Primary Operational Role	Subcategory	Representative Models	Main Use/Characteristics	Boundary Cases/Notes
Generative language models	Direct grading and feedback generation	Closed-source	GPT-4/GPT-4o; Claude; Gemini	Strong generative and explanatory capacity; high agreement with human raters in structured settings	May also support retrieval, embeddings, or auxiliary processing when not used for direct scoring
Generative language models	Direct grading and feedback generation	Open-source	LLaMA; Mistral/Mixtral; Falcon; Qwen	Local deployment, institutional control, and domain adaptation	Role depends on deployment design; may act as grader or support component
Encoder-based transformers	Semantic similarity, representation, and retrieval	Specialized encoders	SBERT; SciBERT; CodeBERT; Longformer/BigBird	Semantic matching, representation, retrieval, and auxiliary scoring support	When fine-tuned for score prediction, they operate as graders
Text-to-text models	Instruction-oriented transformation in formalized assessment tasks	Representative examples	T5; Flan-T5; UL2	Structured prompting, reformulation, transformation, and few-shot grading scenarios	Intermediate role between encoding and generation; useful in hybrid workflows

Note: This taxonomy is role-based rather than architecture-deterministic. The same model family may perform different functions depending on its position in the assessment pipeline. For example, encoder models fine-tuned for score prediction may act as graders, while generative models used only for retrieval or embeddings may function as semantic tools.

The role-based classification reveals that assessment failures are not solely model-dependent but role-dependent. Consequently, educational deployment cannot rely on a

single undifferentiated LLM evaluator; instead, it requires a controlled orchestration in which analytical, grounding, verification, and reporting functions are explicitly constrained.

2.1. Generative Language Models

Generative LLMs are used directly as virtual graders. They receive a task, a rubric, and a student response and on this basis generate a numerical score, argumentation, and feedback. This group includes both closed-source and open-source models that can be used in a local environment.

2.1.1. Closed-Source Generative Models

The GPT family of models (GPT-3.5, GPT-4, GPT-4o, o1) is among the most frequently studied in automated assessment. With clearly structured prompts and analytical rubrics, GPT-4 has achieved levels of agreement with human raters in metrics such as QWK, Cohen's κ , and ICC comparable to inter-rater consistency between experts [1,8]. Similar strong performance has also been reported for models such as Claude and Gemini in the grading of free text and code [24–27].

2.1.2. Open-Source Generative Models

Open-source LLMs such as LLaMA, Mistral/Mixtral, Falcon, and Qwen are particularly relevant in educational contexts where local deployment, data protection, and institutional control are priorities. Their main advantage is that they can be adapted more directly to specific disciplines, languages, and retrieval-based architectures grounded in course materials [10,28].

2.2. Encoder Transformers

Unlike generative models, encoder transformers typically function not as direct graders, but as semantic tools for representation, comparison, and retrieval. This group includes SBERT, SciBERT, CodeBERT, Longformer, and BigBird. Such models extract contextual embeddings that can be used for semantic similarity, retrieval in RAG systems, or as input to downstream classification and regression modules [29,30]. In practice, however, their role is not fixed: when fine-tuned to predict scores directly, encoder-based models may also function operationally as graders rather than only as semantic tools.

2.3. Text-to-Text Models

Text-to-text models such as T5 and Flan-T5 occupy an intermediate position. They are suitable for assessment scenarios that require structured reformulation, transformation, or instruction-following behavior in a unified text-to-text format, especially in few-shot and more formalized tasks [31,32].

2.4. Methodological Significance of the Taxonomy

The proposed taxonomy is not intended merely as a descriptive classification of contemporary model families, but as a methodological instrument for analyzing and designing reliable architectures for automated assessment. Its central assumption is that the performance and failure modes of LLM-based assessment systems are not determined solely by model capability, but are fundamentally shaped by the specific evaluative role assigned to the model within the assessment workflow. In this sense, autonomous evaluators, rubric interpreters, explanatory feedback generators, and multimodal analyzers exhibit distinct patterns of instability, contextual drift, and pedagogical risk depending on their operational function rather than their underlying architecture.

From this perspective, assessment failures should be understood as role-induced rather than model-inherent. The same underlying model may produce substantially different reli-

ability characteristics depending on whether it is used as a direct scoring agent, a semantic feature extractor, a retrieval-support component, or a feedback generation module. Consequently, the taxonomy emphasizes dominant operational roles within a given assessment design, rather than enforcing a rigid mapping between model families and fixed functions. This allows for the existence of hybrid and transitional configurations, in which encoder-based models may be extended toward scoring tasks when fine-tuned for prediction, while generative models may be constrained to auxiliary semantic or retrieval-oriented functions.

Importantly, the taxonomy is not architecture-deterministic and should not be interpreted as a fixed ontology of model capabilities. Instead, it provides a methodological basis for reasoning about how different role assignments shape the behavior of assessment systems and how reliability risks emerge from these assignments. This perspective has direct implications for system design: reliable educational deployment requires not monolithic model usage, but explicitly structured architectures in which model roles are deliberately constrained, externally grounded, and operationally validated through complementary control mechanisms such as rubric constraints, retrieval grounding, and structured output enforcement.

3. Prompt Engineering in Automated Assessment

The effectiveness of large language models as automated assessors depends not only on their architecture, but also on the way in which they are instructed. Prompt engineering has become an important methodological practice through which the role of the model, the context of the task, the assessment criteria, and the format of the output are specified [6,33]. In the context of automated assessment, the literature outlines a clear progression from zero-shot prompting to few-shot strategies, structured reasoning, and rubric-constrained prompting [34]. In the present study, these prompting principles are considered only insofar as they support the design of a controlled and operational assessment framework.

3.1. Evolution of Prompting Strategies

Zero-shot prompting does not include sample scores and is therefore relatively easy to implement, but for complex or highly interpretive tasks it often leads to higher variability in results [35]. Few-shot prompting mitigates this problem by introducing sample response–score pairs, through which the model implicitly learns the boundaries between performance levels [34,35]. For more complex assessment scenarios, structured reasoning techniques such as Chain-of-Thought (CoT) can improve both the accuracy and transparency of the decision [34]. Together, these developments show that LLM-based grading can be organized as a more controlled procedure rather than as unrestricted generation. Figure 1 summarizes this progression from minimal prompting toward more structured and controlled prompting strategies in LLM-based automated assessment.

3.2. Rubric-Oriented Prompting

One of the most effective mechanisms for stabilizing assessment is the integration of analytical rubrics directly into the prompt. Clearly formulated criteria enhance consistency and reduce the risk of arbitrary deviations [6,12,36]. In this sense, the rubric functions both as a pedagogical framework and as an algorithmic constraint on the generative process. Additional stability can be achieved through more detailed rubric definitions and decision structures, while standardized output formats such as JSON make the results easier to process, verify, and integrate into LMS environments [12,28,37]. In implemented educational systems, such prompt constraints play a central role in transforming LLM assessment from free-form generation into a controlled scoring procedure.

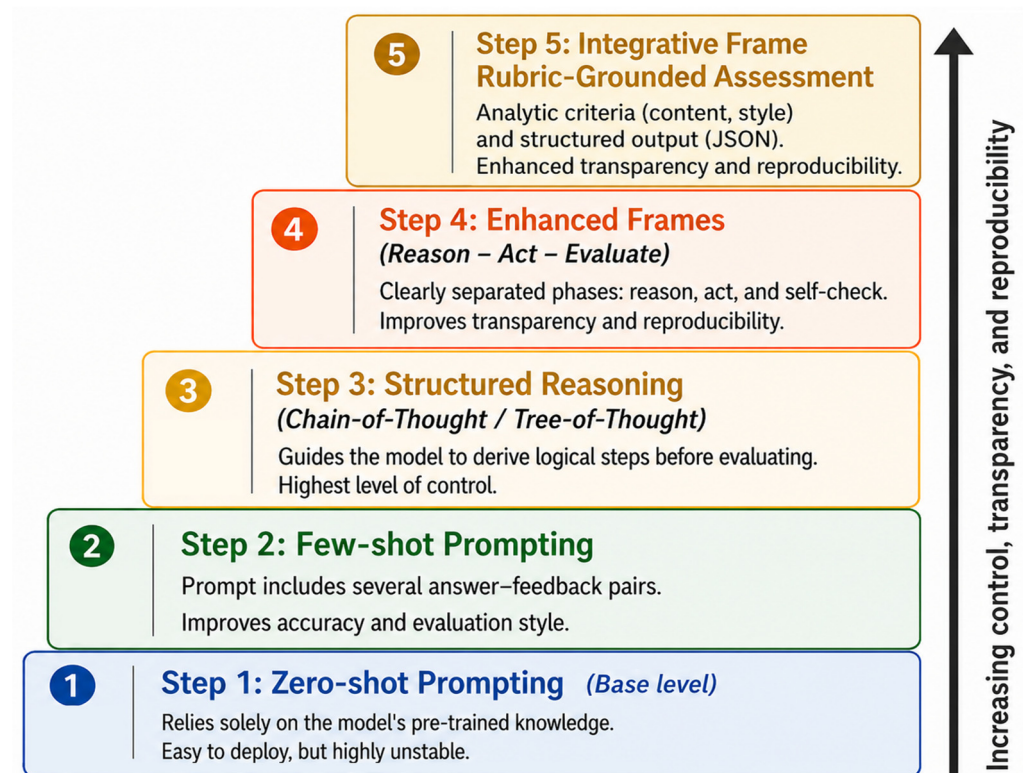


Figure 1. Evolution of prompting strategies in LLM-based automated assessment.

3.3. Generation Parameters and Stability

The reliability of LLM-based assessment also depends on generation parameters. Low temperature values tend to produce more deterministic behavior and reduce variability between model invocations [6]. In more stochastic settings, robustness may be improved through repeated calls and aggregation strategies such as majority voting [34].

3.4. Limitations of Prompt-Based Assessment

Despite its effectiveness, prompt engineering has important limitations. Prompt-based assessment remains sensitive to small textual variations, and even minor changes in wording can lead to differences in scores, especially when criteria are not clearly defined [6,35]. Moreover, prompt engineering alone does not solve the problem of contextual grounding. Without access to verifiable learning materials, the model may generate plausible but inaccurate interpretations, i.e., hallucinations [7,10]. Prompt engineering is therefore a necessary but not sufficient condition for reliable automated assessment. It must be complemented by an architectural mechanism that constrains assessment within a specific and verifiable learning context. In the present framework, this role is fulfilled by RAG.

4. Limitations of Standalone LLM Graders

Despite their high performance and the often reported agreement with human experts, large language models cannot be considered as fully autonomous and reliable graders. The literature consistently highlights limitations related to the validity, stability, interpretability, and contextual adequacy of automated grading [7,10].

4.1. Validity and Hallucinations

One of the most serious risks is the phenomenon of hallucination—the generation of convincing-sounding but factually inaccurate or unrelated statements. In the grading context, this can mean attributing arguments or concepts that are actually absent from

the student's text [9,33]. Such biases undermine the construct validity of grading, as the result is based on a statistically plausible interpretation rather than the actual content of the response. When there is no mechanism for grounding grading in specific learning material, the risk of such errors increases.

4.2. Instability and Reproducibility

LLMs function as probabilistic models and can generate different grading outcomes for the same input. Even at low temperatures, complete repeatability is not guaranteed [7,33]. An additional difficulty arises when updating models, which can change their estimation profile without an explicit change in methodology [12]. This calls into question the long-term comparability of results, especially in contexts of formal and high-stakes assessment.

4.3. Systematic Discrepancies and Biases

Empirical studies have shown that different LLMs can exhibit systematic rigor or, conversely, a tendency to assign higher scores than human raters [25,27]. A “central tendency” effect is also observed, in which final scores tend to cluster around the mean [38]. Models can also reproduce biases inherent in the training data [39], as well as be influenced by surface features of the text—for example, by lengthening the response or including terms that statistically correlate with a higher score [40].

4.4. Interpretability and Credibility of Explanations

A significant limitation of LLM-based grading is related to the interpretability of the generated explanations. Although the model can formulate a reasoned textual justification, this does not necessarily mean that the explanation reflects the real grounds on which the grade was assigned. Therefore, the plausibility of the explanation is not a sufficient guarantee of its credibility. In an educational context, this limits the use of such explanations as a fully reliable basis for pedagogical interpretation. Interpretability in LLM-based grading should therefore not be seen as an automatic property, but as a characteristic that needs to be reinforced through analytical rubrics, structured output, and contextual constraints on the grading process.

4.5. Lack of Domain-Specific Grounding and Infrastructural Limitations

Generic LLMs are not trained on specific curricula or standards. As a result, grading is often based on general linguistic similarities rather than on checking for compliance with specific learning material [2,41]. This creates particular difficulties in disciplines with strictly defined terms and formal structures. In addition, the use of cloud models raises issues related to the confidentiality of student data and compliance with institutional requirements [7].

4.6. Synthesis of Limitations

As stand-alone graders, LLMs demonstrate four main limitations:

1. Lack of contextual grounding;
2. Probabilistic instability;
3. Limited interpretability;
4. Risk of bias and manipulation.

These limitations show that stand-alone LLM graders should not be treated as fully autonomous and institutionally reliable assessment agents. Their effective educational use requires a broader architectural framework that provides contextual grounding, structured criteria, and verifiable outputs. In the present study, this need is addressed through a

developed and integrated system architecture in which these control mechanisms are operationally implemented.

5. Architectural Strategies for Controlled Assessment

Integrating LLM into automated assessment requires that the model be not only powerful, but also controlled, constrained, and methodologically sound, so that its behavior is reproducible, robust, and consistent with academic assessment standards. In the developed and implemented system, this is achieved by combining several interrelated mechanisms: role prompting, contextual grounding via RAG, structured machine-readable outputs, formalized rubrics, and workflow orchestration in an LMS environment. At a practical level, these principles are implemented through an integrated architecture based on Moodle Web Services, a vector database for semantic retrieval, and an n8n workflow that coordinates the entire assessment process in real time. It is in such a technological context that control over the LLM ceases to be an abstract idea and becomes a verifiable, traceable, and pedagogically sound mechanism.

The proposed framework for controlled educational integration of LLMs is expressed in this section through six interrelated architectural components: role prompting, RAG-based grounding, structured machine-readable outputs, formalized rubrics, workflow orchestration, and institutional LMS integration. Taken together, these components define a reproducible design logic through which LLM-based grading can be constrained, traced, and embedded in a real educational environment.

5.1. Role Prompting as a Behavioral Constraint Mechanism

In automated assessment, role prompting is not limited to a single instruction, but acts as a behavioral regulation mechanism for the model throughout the entire grading process. The model is explicitly assigned the role of an automated grader, not a free generative system. Thus, its task is not to “add” content, but to analyze the student response against predefined criteria and permissible context.

In a production LMS architecture, this principle is implemented through an orchestration layer that submits to the model a strictly structured request containing the student response, the rubric, and the extracted learning context. The instructions limit the model to referring to only three sources: the specific student response, the rubric defined by the instructor, and the relevant context extracted from the learning materials. The use of external knowledge is explicitly prohibited by the system prompt. As a result, the LLM functions not as a source of new content, but as a tool for classification and interpretation within a predefined pedagogical logic. An additional advantage of this approach is that standardized input limits random variations between individual performances. When the model is consistently fed the same kinds of information—question wording, student response, rubric, relevant context, and experience identifier—conditions are created for higher reliability and reproducibility.

5.2. Limiting Hallucinations Through RAG and Evidence-Constrained Input

One of the most significant weaknesses of LLMs as independent graders is the risk of hallucinations and the use of irrelevant general knowledge. In the developed and implemented framework, this risk is limited by evidence-constrained evaluation, implemented through RAG. Instead of the model evaluating responses based on its general pre-trained representations, it works with an extracted, verifiable, and course-specific context. In practice, this means that before grading, the system first extracts relevant contextual passages from the learning materials, and then feeds them to the model in clearly demarcated sections, for example:

```

<context> ... </context>
<answer> ... </answer>
<rubric> ... </rubric>

```

This organization of the input is not just a technical convenience, but also a means of disciplining the grading process. Through it, the model works within the framework of clearly demarcated and verifiable information sources. The instructions explicitly prohibit the use of information outside the content included in the <context> section. In this way, grading is not based on the “general knowledge” of the model, but on the specific learning material provided by the teacher.

The architecture also allows for an additional layer of verification. If necessary, the model can be asked to indicate which contextual passages support the choice of level for each criterion. If sufficient evidence is lacking, the system can impose a conservative regime and award a lower level. This approach does not completely eliminate the possibility of errors, but it significantly reduces the frequency and severity of hallucinations and increases the correspondence between grading and the material taught.

5.3. JSON as a Verification and Traceability Mechanism

Structuring the grading result in JSON format is a key methodological component of the system. In this context, JSON is not just a data exchange format, but a formal protocol that defines the mandatory elements of grading and the relationships between them. It describes the criteria, the selected levels, the overall score, the evidence used, and a brief justification. This structure performs several important functions. First, it allows for automatic verification that all criteria in the rubric have been processed. Second, it forces the model to follow a fixed format, thus limiting the deviation towards free and difficult-to-verify generation. Third, it facilitates the recording and logging of results in the LMS environment, since each component of grading can be stored, tracked, and analyzed separately.

At a conceptual level, JSON functions as a mechanism for traceability and scientific verifiability. It allows for subsequent inspection of grading results, comparison between different implementations of the model, and checks for internal consistency—for example, whether the sum of the criterion scores matches the final score. Furthermore, the formalized output creates conditions for analyzing consistency at the level of an individual criterion, which is essential for validating the reliability of the system.

5.4. Rubric as a Formal Pedagogical Constraint

Rubrics represent the formalized pedagogical model of assessment in a controlled LLM-based system. In LMS environments such as Moodle, they can be stored as structured objects associated with assignments through the Grading method = Rubric mechanism. When programmatically retrieved, the rubric can be transformed into a machine-processable JSON format that can be fed to the AI agent as a formal assessment framework. After this transformation, the rubric contains the assessment criteria, the performance levels for each criterion, the corresponding points, and the textual definitions of the levels. In this way, it functions as a strict formal model that limits the output of the language model to the predefined criteria and allowable levels. Instead of generating a free and potentially variable grading, the model selects a specific level for each criterion within the structure defined by the teacher.

This makes the rubric a fundamental mechanism for pedagogical control. Through it, grading remains compatible with the standard Moodle mechanism and at the same time acquires higher objectivity, comparability, and traceability. The practical implementation of

an analytical rubric in Moodle is presented in Figure 2, where both the general criteria and the detailed levels with the point scale are shown.

Grading Criteria

Task 1: User Story (20 points)

Level	Description	Points
Excellent	Complete "As a... I want... So that..." format, 4+ clear and testable Acceptance Criteria	17–20
Good	Partial format (missing "so that"), 2–3 Acceptance Criteria	10–16
Poor	Wrong format, less than 2 criteria or not testable	0–9

Task 2: Requirements Techniques (15 points)

Level	Description	Points
Excellent	Correct technique for all 3 cases + clear explanation why	13–15
Good	2 out of 3 correct answers with partial explanations	7–12
Poor	1 or 0 correct answers, missing explanations	0–6

Correct answers: (a) Surveys, (b) Surveys, (c) Individual interviews

(a)

Task 1: User Story Format	Missing "As a/I want/So that" format 0 points	Partial format (1-2 parts) 3 points	Complete format with all 3 parts 5 points
Task 1: User Story Completeness	Role/goal/benefit unclear 0 points	Role and goal clear, benefit weak 3 points	All parts clear and relevant 5 points
Task 1: AC Quantity	0-1 acceptance criteria 0 points	2-3 acceptance criteria 3 points	4+ acceptance criteria 5 points
Task 1: AC Quality	Not testable or irrelevant 0 points	Some testable, some vague 3 points	All testable, specific, relevant 5 points

(b)

Figure 2. Implementation of an analytical rubric in Moodle: (a) general criteria; (b) detailed levels and point scale.

In the empirical study reported in Section 6, instructor-defined Moodle rubrics are used in both manual grading and an AI-based workflow, allowing for direct comparison at the criterion level within the same formal grading structure.

Although the illustrated rubrics are drawn from software engineering tasks, the framework is not limited to this disciplinary context. In the implemented architecture, the analytical rubric is defined by the instructor for each specific assessment activity, while the contextual grounding layer retrieves course-specific materials automatically according to the corresponding course and task. Thus, the domain-specific elements of the system are the rubric content and instructional knowledge base, whereas the underlying architectural logic remains transferable across disciplines. At the same time, the present empirical validation is limited to a single course context, and broader cross-disciplinary testing remains future work.

5.5. Fairness, Neutrality, and Reliability

The integration of LLM systems into educational assessment inevitably raises the question of fairness, neutrality, and reliability of grading itself and the generated feedback. Within the proposed architectural logic, these principles are not considered secondary technical features, but rather basic methodological requirements.

Fairness implies grading according to uniform and transparent rules, without the influence of irrelevant factors such as writing style, random linguistic variation, or individual features of expression. In the system, this is achieved by limiting the analysis to the context extracted through RAG and the predefined rubric. All learners are graded according to identical criteria, with a standardized input format and unified instructions for the model. Similar observations are also found in the literature, where [2] report high consistency of GPT-4 at different cognitive levels, although with a tendency towards higher scores compared to human raters.

Neutrality is achieved through a combination of role prompting, a strictly structured input, and a machine-readable output. Limiting the model to a specific <context> reduces the risk of carrying over social, cultural, or linguistic biases that can be inherited from the broad LLM training corpora. This keeps the focus on the content of the response and its correspondence to the learning material and rubric.

Reliability can also be viewed through the lens of interrater agreement. While variability in manual scoring often stems from subjective interpretations, systems working with a stable rubric and standardized input demonstrate higher internal consistency. Studies on the ASAP corpus show that transformer-based models achieve Quadratic Weighted Kappa (QWK) in the range of 0.6–0.8, which according to [42] corresponds to significant to almost complete agreement; similar values have been reported by [43,44]. This shows that, given clearly defined instructions and context, automated grading can reproduce a level of reliability comparable to that of expert graders.

Beyond fairness, neutrality, and reliability, the use of LLMs in automated assessment also raises broader ethical questions related to data protection, procedural transparency, and student learning. In the implemented framework, several practical safeguards were introduced to reduce unnecessary exposure of personal information: the system is deployed in an isolated containerized environment, communication between Moodle and the orchestration layer is performed through token-based Web Services, and access is restricted through a dedicated service account with limited permissions. In the experimental analysis, student records were additionally anonymized through unique identifiers that do not allow direct participant identification. At the same time, the framework does not claim full interpretability of the underlying language model. Instead, it aims at procedural transparency through rubric-constrained prompting, contextual grounding in course materials via RAG, and structured machine-readable outputs that support logging, traceability, and post hoc inspection of grading decisions. From a pedagogical perspective, the system is intended to support timely, criterion-based feedback and more consistent assessment, but its educational value depends on careful institutional use. If deployed uncritically, automated grading may shift attention toward rubric compliance rather than deeper reasoning and development. For this reason, the framework should be understood not as a replacement for pedagogical responsibility, but as a controlled institutional tool whose use requires explicit safeguards, transparency, and continued human oversight.

5.6. Architectural Implementation of Supervised Grading

The principles described above do not remain only at a conceptual level, but are realized in a concrete technical architecture developed and integrated by the authors. The implemented system is organized as a multi-layer architecture that combines Moodle

as the LMS layer, n8n as the orchestration layer, Supabase as the semantic vector layer, OpenAI models as the AI layer, and MariaDB as the operational database layer of Moodle. This architecture enables the grading process to be executed as a closed, traceable, and institutionally integrated workflow rather than as an isolated model invocation.

At the entry point of the system, Moodle functions as the educational platform through which students access course materials, submit their assignments, and receive grades. In addition to its standard LMS role, Moodle stores the assignment definitions, the analytical rubrics, and the submission metadata required for automated grading. The connection with the AI layer is realized through two complementary mechanisms: a webhook plugin that sends submission events to n8n in real time, and Moodle Web Services, which expose the necessary REST-based API functions for retrieving course, module, rubric, and grading data and for writing the final grade back into the LMS.

The orchestration layer is implemented in n8n, which coordinates the entire grading lifecycle through three interconnected workflows. The first workflow is responsible for periodic course synchronization. It retrieves Moodle credentials and the current list of courses through Web Services and ensures that the AI layer operates on up-to-date course information. The second workflow implements the document processing and RAG pipeline. It downloads course resources from Moodle in formats such as PDF, DOC/DOCX, and PPT/PPTX, extracts their textual content, transforms the extracted content into embeddings, and stores the resulting semantic representations in Supabase. The third workflow performs the actual AI-based grading. It receives the student submission through the webhook mechanism, retrieves the relevant rubric and assignment metadata from Moodle, searches the vector database for contextually relevant material, invokes the grading model with the rubric, the student answer, and the retrieved context, and finally sends the grading result back to Moodle.

The semantic layer is implemented through Supabase with the pgvector extension and functions as the retrieval component of the RAG architecture. Its central role is to store chunked learning materials together with their embedding representations and metadata, including the course identifier, filename, modification date, and document version. In practical terms, this layer is used to retrieve top-k contextually relevant passages from course materials before grading. The RAG pipeline was configured with a Recursive Character Text Splitter using segments of 1000 characters with an overlap of 200 characters in order to preserve local semantic continuity during retrieval. Vector representations were generated with the OpenAI text-embedding-3-small model (1536 dimensions) and stored in Supabase PostgreSQL with the pgvector extension, using an ivfflat index with cosine-based similarity search. During retrieval, the system applies course-level filtering and, when available, an additional file-level constraint derived automatically from the assignment context. Candidate passages are ranked by cosine similarity, filtered by a relevance threshold of 0.70, and passed to the grading layer through a top-k strategy, typically using 3–5 retrieved segments. Retrieval quality is controlled through this combination of overlap-aware chunking, cosine-based ranking, relevance thresholding, contextual isolation by course and resource, and re-vectorization of newly added or modified learning materials. This makes it possible to constrain the language model to verifiable instructional content rather than allowing it to rely on unrestricted background knowledge. In addition to the main document table, the semantic layer also supports synchronization and analytical logging through auxiliary tables for newly detected files and for storing grading-related results.

The AI layer is implemented with OpenAI models serving two distinct but connected roles. The embedding model is used to vectorize learning materials and search queries for semantic retrieval, while GPT-4.1-mini is used as the grading model for rubric-based evaluation of the student response. In this configuration, the model does not operate as

an autonomous free generator. GPT-4.1-mini was selected as the grading model because it provided a suitable balance between instruction following, structured output reliability, multimodal capability, response latency, and cost for integration in a real Moodle-connected assessment workflow. The choice was therefore driven primarily by deployment-oriented considerations rather than by an attempt to identify a universally optimal model. Larger proprietary alternatives such as GPT-4o and Claude were considered during system design, but were less attractive under the practical cost and integration constraints of the implemented architecture, while local open-source alternatives remained limited by the available hardware, especially for multimodal processing and stable machine-readable output. Importantly, the architecture itself remains model-agnostic, which means that the contribution of the study lies in the controlled assessment framework and its institutional integration rather than in dependence on a single specific model. Instead, it is embedded in a tightly controlled process in which the input consists of the student submission, the formal rubric structure, and the course-specific context extracted through RAG. The output is constrained to a structured machine-readable format that supports verification, logging, and LMS reintegration.

At the data level, MariaDB continues to serve as the operational database of Moodle, storing institutional information such as users, courses, assignments, submissions, and grades. The relation between MariaDB and the rest of the architecture is indirect but essential: Moodle manages the academic workflow and persists the official grading records, while n8n and Supabase extend Moodle with semantic retrieval and automated assessment capabilities. Thus, the implemented architecture preserves compatibility with the standard LMS infrastructure while adding an external but fully connected AI-based grading layer.

Figure 3 summarizes the implemented architecture and its principal data flows. Path (1) represents the real-time submission webhook from Moodle to n8n, while path (2) represents REST-based retrieval of course files, rubrics, and metadata required for synchronization and grading. Paths (3) and (5) capture the interaction between n8n and the OpenAI layer for embedding generation and rubric-based grading, whereas path (4) represents the background indexing and retrieval cycle between n8n and Supabase within the RAG pipeline. Path (6) closes the operational loop by returning the resulting grade and feedback to Moodle, while path (7) indicates Moodle's connection to its operational MariaDB layer for core LMS data. In this way, Figure 3 presents not a hypothetical design, but the actual operational architecture through which the proposed framework is realized, including the three connected workflows for course synchronization, document indexing and RAG preparation, and AI-based grading.

To validate the applicability of the proposed architecture beyond purely textual grading, the implemented system was extended with a multimodal assessment component. This extension enables the same controlled architectural logic to be applied to visual artifacts submitted by students, such as UML diagrams, while preserving compatibility with rubric-based grading, JSON-structured outputs, and LMS reintegration.

5.7. Multimodal Extension: Visual Assessment via Vision Model

5.7.1. Visual Assessment Object and Criteria

In addition to text responses, the implemented architecture also supports automated assessment of visual artifacts submitted as image-based assignment materials, including UML diagrams as well as other kinds of images relevant to the task. This functionality extends the applicability of the system to practical tasks that cannot be graded solely through text analysis. Operationally, the multimodal branch is designed to process attached visual files provided by students as part of their submission, rather than being limited to a single diagram type.

Visual assessment is implemented through the Vision capabilities of GPT-4.1-mini, which analyzes the image against predefined criteria in the rubric. In the configuration used, the UML diagram rubric includes criteria related to states, transitions, guard conditions, business rules, and correctness of UML notation.

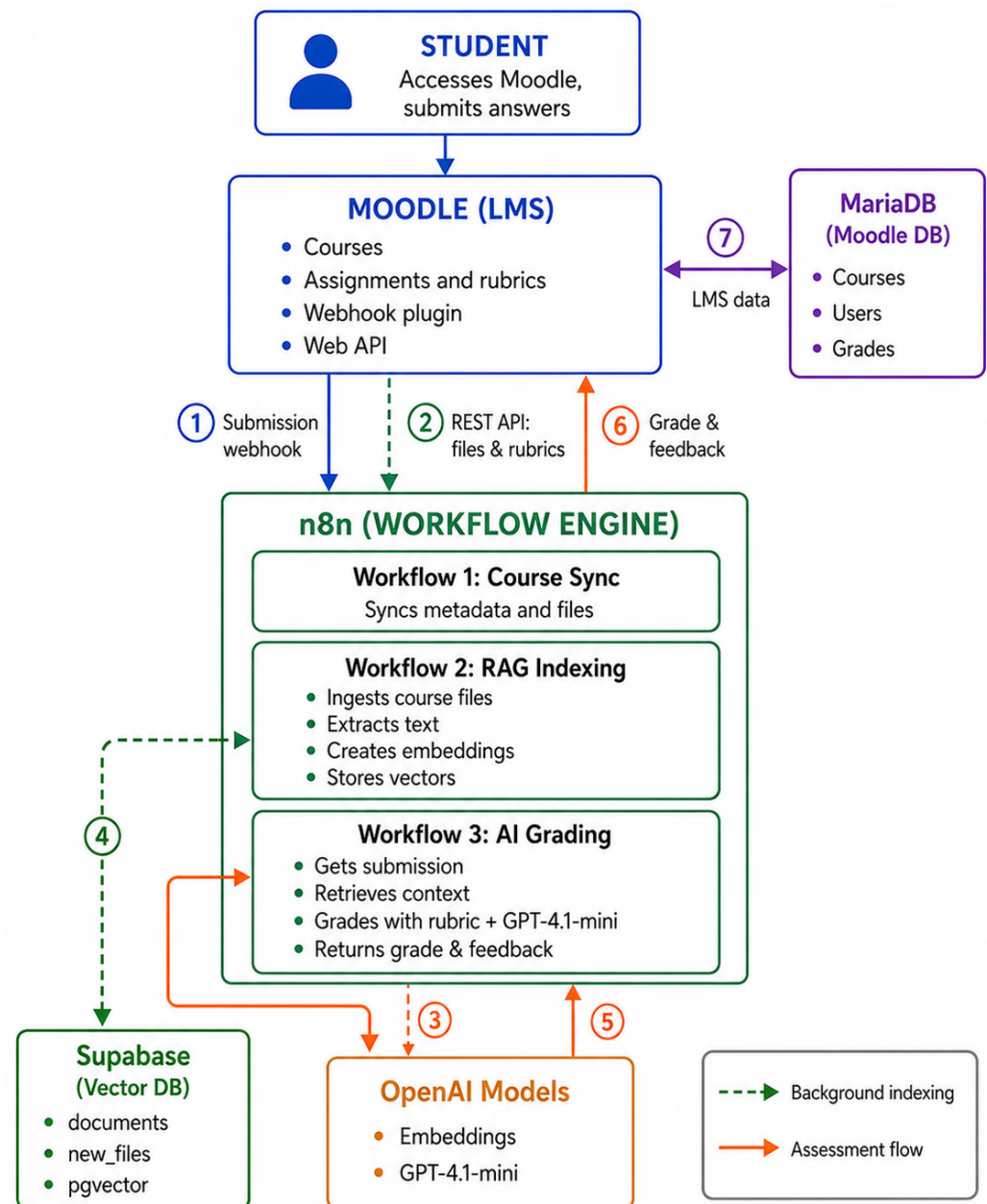


Figure 3. Integrated architecture and operational data flows of the proposed Moodle-connected assessment framework.

Grading is performed using an analytical rubric consisting of six criteria, with each criterion containing three performance levels with fixed values of 0, 3, or 5 points. The criteria cover the main structural and semantic elements of the diagram, including:

- Presence of all mandatory states (Draft, Submitted, Under Review, Approved, Rejected, Cancelled, and Expired);
- Correct use of initial and final states;
- Presence of correct transitions between states;
- Use of conditional transitions (guard conditions);

- Correct application of business rules;
- Correctness of UML notation.

The criteria and levels of this visual rubric are summarized in Figure 4.

Task 5: States Present	No diagram submitted OR 0-2 states 0 points	3-5 states 3 points	All 7 states present 5 points
Task 5: Initial/Final States	No diagram submitted OR Missing both 0 points	One present 3 points	Both with correct notation 5 points
Task 5: Basic Transitions	No diagram submitted OR 0-3 transitions 0 points	4-6 transitions 3 points	7+ transitions 5 points
Task 5: Guard Conditions	No diagram submitted OR No guards 0 points	Some guards 3 points	All rules as guards 5 points
Task 5: Edit/Withdraw Rules	No diagram submitted OR Not implemented 0 points	Partially correct 3 points	Fully correct 5 points
Task 5: UML Notation	No diagram submitted OR Not UML 0 points	Partial UML notation 3 points	Correct UML state diagram notation 5 points

Figure 4. Rubric for visual assessment of the UML State Diagram—criteria and levels (0/3/5 points).

The analysis performed by the Vision model allows the automatic identification of visual elements and the grading of their correctness against the requirements of the assignment. The obtained results are then integrated into the overall rubric structure, ensuring consistency between textual and visual grading.

5.7.2. Architectural Implementation of Vision Analysis

Architecturally, Vision analysis is implemented as an integrated conditionally activated branch in the main workflow for automated grading on the n8n platform. When the system detects the presence of an attached file, a separate process for visual analysis is automatically triggered. The logic of the conditionally activated Vision branch is illustrated in Figure 5.

After performing the visual analysis, the results obtained are integrated into the overall rubric structure so that the final grade reflects both the textual and visual components of the student's response. In this way, multimodal grading is not a side addition, but an organic part of the architecture for controlled automated assessment. It allows for a more complete coverage of practical and modeling skills, without violating compatibility with the standard Moodle grading mechanism.

While the proposed framework is theoretically derived from the reliability deficiencies associated with uncontrolled LLM deployment, its practical educational relevance depends on whether the implemented workflow can produce reliable, traceable, and pedagogically usable results under authentic grading conditions. For this reason, the following section

reports an empirical validation of the Moodle-connected framework against independent expert grading, with additional exploratory analysis of the UML-based multimodal branch.

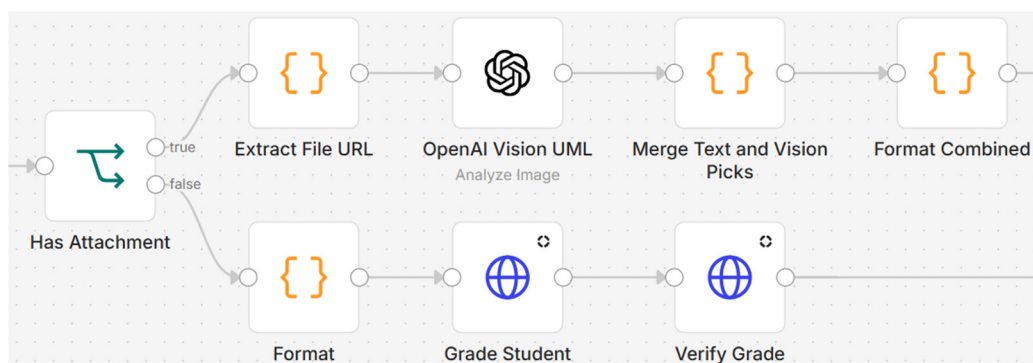


Figure 5. Conditionally activated multimodal assessment branch in the deployed n8n workflow environment.

6. Methodological Design and Empirical Validation

The empirical study was designed to evaluate the implemented controlled LLM-based assessment framework under realistic educational conditions. The primary objective was to examine whether the Moodle-connected, rubric-constrained, and RAG-grounded workflow can produce assessment outcomes aligned with independent expert grading, support multimodal UML assessment, and generate structured feedback suitable for LMS reintegration. The reported empirical results aim to assess the feasibility of the proposed framework rather than to establish competitive benchmark comparisons.

The evaluation focused on three dimensions: aggregate agreement with expert grading, exploratory validation of the UML-based multimodal branch, and expert assessment of feedback quality. Performance was measured using agreement, reliability, correlation, error, and bias indicators, together with criterion-level analysis for the UML task. The results are summarized in Tables 2–5.

Table 2. Mapping of Task 5 UML rubric criteria to Bloom’s cognitive levels.

UML Subcriterion	Bloom Level
Present states	Remember
Initial/Final states	Apply
Basic transitions	Apply
Guard conditions	Analyze
Edit/Withdraw rules	Analyze
UML notation	Create

Table 3. Overall validation results for Task 5 (UML visual assessment).

Metric	All Task 5 Cases	Submitted UML Diagrams Only
Number of students	32	24
Mean Expert raw score	4.234	4.979
Mean AI raw score	4.181	4.908
Exact agreement (rounded grade, %)	62.5	50.0
MAE	0.158	0.211
RMSE	0.298	0.344
Bias (AI–Expert)	−0.053	−0.071

Table 4. Criterion-level agreement between expert and AI for the UML assessment task (submitted diagrams only).

UML Criterion	N	Mean Expert	Mean AI	Exact Agreement (%)	MAE
Basic transitions	24	4.542	4.542	91.67	0.167
Edit/Withdraw rules	24	3.458	3.000	83.33	0.458
Guard conditions	24	1.292	1.500	87.50	0.458
Initial/Final states	24	4.375	4.250	91.67	0.292
Present states	24	4.583	4.500	87.50	0.250
UML notation	24	4.167	4.333	91.67	0.167

Table 5. Quality of AI-generated feedback for the UML task (submitted diagrams only).

Feedback Quality Dimension	Mean	SD	N
Clarity	4.472	0.506	24
Usefulness	4.472	0.506	24
Correctness	4.500	0.507	24

6.1. Methodological Design and Validation Scope

The empirical validation was conducted in a real university e-learning environment using a dedicated Moodle platform configured for the purposes of the study. The experiment took place during a remedial examination session in the undergraduate course *Systems Engineering* and involved two parallel course sections with equivalent content, structure, tasks, and grading criteria: one delivered in Bulgarian and one in English. This design made it possible to examine the behavior of the proposed system under language variation while keeping the pedagogical and assessment model unchanged. The examination was carried out in a controlled on-site computer laboratory using university-owned machines with restricted access to external online resources. Students were explicitly instructed not to use external AI tools, and the session was supervised by the teaching staff, so that the collected responses reflected the students' own knowledge and reasoning.

The dataset comprised 32 students in total, including 13 in the Bulgarian section and 19 in the English section. Each student completed an exam consisting of five open-ended tasks, resulting in 160 task-level evaluation instances. The tasks covered structured textual, analytical, argumentative, and modeling-oriented responses, including a UML state diagram task. All responses were graded automatically by the proposed AI-based system using the same analytical rubrics that were independently applied by a human expert for reference evaluation. The AI-generated scores were used exclusively for scientific analysis, whereas the official institutional grades remained those assigned by the instructor in accordance with university procedures. The evaluation design was therefore aimed at validating the system in terms of agreement with expert grading, behavior across Bloom-related cognitive levels, and feedback quality, with additional focused analysis of the UML-based multimodal branch.

The analytical rubrics used in the study were defined by the course instructor and implemented in Moodle through the Assignment → Advanced grading configuration. The same rubric definitions were applied in both the instructor's manual grading and the AI-based workflow, allowing criterion-level comparison within a shared formal assessment structure.

The human reference evaluation was provided by the responsible course instructor, who uses these rubric criteria in the regular assessment of the course.

The methodological design was structured to examine this research question through the implemented system itself, by documenting the proposed architectural approach and by validating its behavior in a defined university setting.

6.2. Aggregate Validation of the Implemented Grading System

The implemented system was evaluated against an independent human expert using a compact set of complementary metrics capturing agreement, reliability, correlation, error magnitude, and systematic deviation. The analysis was conducted on the full dataset of 160 task-level observations.

At the aggregate level, the system demonstrated high agreement with the expert evaluation. For the full dataset, the obtained results were QWK = 0.806, ICC = 0.868, Pearson's $r = 0.836$, MAE = 0.453, RMSE = 0.810, and bias = -0.011 . Taken together, these results suggest that the proposed methodology approximates expert grading with strong agreement, high reliability, and relatively low numerical deviation within the tested educational setting. The observed bias remained close to zero, indicating no substantial systematic over-scoring or under-scoring by the AI system.

6.3. Exploratory Validation of the UML Multimodal Branch

To examine the multimodal branch beyond purely textual grading, Task 5 was used as a UML-based visual assessment task. Because the UML task required not only visual recognition of diagram elements but also interpretation of their functional and logical role, its rubric criteria were additionally mapped to Bloom's cognitive levels. This mapping was used to describe the cognitive profile of the visual assessment task and to distinguish between criteria based mainly on recognition or application and criteria requiring analysis or creation. Table 2 presents the correspondence between the UML rubric subcriteria and the associated Bloom levels.

At the aggregate level, the system showed close alignment with the expert scores. Across all 32 cases, the mean expert raw score was 4.234, compared with 4.181 for the AI system, with a small negative bias of -0.053 , indicating slightly more conservative AI grading. When only submitted UML diagrams were considered ($n = 24$), the mean expert and AI raw scores were 4.979 and 4.908, respectively, with MAE = 0.211 and RMSE = 0.344. Eight cases corresponded to missing UML diagrams, and these were handled consistently by both the expert and the AI system. The overall results for the UML task are summarized in Table 3.

Criterion-level agreement is summarized in Table 4. Criterion-level analysis showed that the multimodal branch performed best on visually explicit UML elements. Exact agreement reached 91.67% for basic transitions, initial/final states, and UML notation. Lower agreement was observed for present states (87.50%), guard conditions (87.50%), and especially Edit/Withdraw rules (83.33%). The highest MAE values were found for guard conditions and Edit/Withdraw rules (0.458 each), suggesting that the most challenging aspects of visual grading were those requiring interpretation of conditional logic and domain-specific business rules rather than recognition of explicit structural diagram elements.

At the criterion level, the main discrepancies were concentrated in semantically constrained elements rather than in explicit visual structure. Basic transitions, initial/final states, and UML notation were recognized most consistently, whereas present states occasionally showed mismatches related to incomplete state coverage. The most challenging criteria were guard conditions and Edit/Withdraw rules, where the system had to interpret conditional logic and business-rule semantics beyond straightforward diagram structure.

These findings suggest that the main failure modes of the multimodal branch are related less to basic visual detection and more to the interpretation of conditional and rule-dependent diagram elements. The feedback quality results are summarized in Table 5.

The generated feedback for the UML task achieved mean scores of 4.472 for clarity, 4.472 for usefulness, and 4.500 for correctness in cases with submitted UML diagrams. These results provide initial evidence that the multimodal branch was able not only to assign scores, but also to generate feedback that was generally clear, useful, and correct under the applied evaluation scheme. Subcriterion-level feedback quality was highest for initial/final states and UML notation, while somewhat lower but still positive values were observed for guard conditions and basic transitions, suggesting that feedback generation was strongest for explicit structural elements and more limited for criteria involving conditional or rule-based interpretation.

6.4. Summary of Empirical Results

The empirical results provide initial validation evidence that the proposed controlled assessment workflow can approximate expert grading behavior with substantial reliability across both textual and multimodal open-ended tasks. At the aggregate level, the implemented system demonstrated high agreement with human scoring, low numerical deviation, strong rank-order consistency, and minimal systematic bias. In parallel, the multimodal UML branch showed close alignment with expert evaluation not only in score assignment, but also in the generation of clear, useful, and largely correct formative feedback.

A consistent pattern across the reported analyses is that system performance was strongest when the assessment criteria were explicitly formalized, structurally grounded, and directly linked to the analytical rubric. Both the aggregate textual grading results and the UML criterion-level findings suggest that the proposed workflow performs most robustly under conditions where evaluative expectations can be operationalized through structured prompting, retrieval support, and machine-readable score formalization.

Collectively, these findings indicate that the implemented framework is capable of functioning not merely as an isolated grading model, but as a reproducible institutionally deployable assessment pipeline whose outputs remain sufficiently stable, interpretable, and pedagogically usable within the tested university setting.

7. Discussion

7.1. Interpretation of Experimental Results

The results reported in Section 6 provide initial support for the proposed framework as a controlled and institutionally integrated approach to automated assessment. Within the tested university setting, the implemented workflow showed strong alignment with expert grading, low numerical deviation, strong rank-order consistency, and minimal systematic bias. These findings should be interpreted as evidence of implementation-level viability rather than as a broad benchmark of LLM grading accuracy across domains.

The results also suggest that reliable LLM-based assessment depends not only on the selected model, but on the surrounding control architecture. In the proposed framework, the model's evaluative behavior is constrained by the analytical rubric, grounded in retrieved course-specific materials, formalized through structured machine-readable outputs, and embedded in an LMS-based workflow. This combination supports traceability and reduces the risk of uncontrolled free-form grading.

The multimodal UML results add an important qualification to this interpretation. The branch performed most consistently on visually explicit and structurally formalized criteria, such as transitions, initial/final states, and UML notation. Lower consistency was observed for criteria requiring interpretation of conditional logic, domain semantics, or business-rule

dependencies. This suggests that multimodal LLM-based assessment is more reliable when the expected visual and semantic features are clearly operationalized in the rubric.

Overall, the findings support the broader methodological claim of the study: reliable educational deployment of LLMs requires controlled architecture rather than monolithic unrestricted model use. The contribution of the framework lies in the orchestration of control mechanisms—rubric alignment, retrieval grounding, structured output, and LMS integration—rather than in unrestricted model autonomy.

7.2. Comparative Positioning Against Existing Integrated Assessment Systems and Architectures

This section discusses a comparative positioning of the proposed framework relative to existing integrated assessment systems and competing architectures from four categories: Moodle-based AI integrations, commercial grading infrastructures, AI-assisted integrity tools, and RAG-based grading systems. The comparison focuses on architectural capabilities rather than implementation details alone.

Although Moodle-AI integrations have been reported in the literature, most existing systems implement only partial functionality relative to the proposed framework. The closest comparator is the Moodle-integrated assistant [45], which enables semantic assessment and adaptive feedback within Moodle, but relies on fine-tuned transformer-based models and tutoring-oriented adaptation rather than RAG-grounded LLM grading [45]. Similarly, plugin-based and Moodle-connected solutions such as [46,47] demonstrate automated feedback generation and domain-specific grading within LMS environments, but are primarily based on deterministic or rule-based mechanisms and do not combine rubric-constrained LLM evaluation, retrieval grounding, structured machine-readable outputs, and multimodal assessment within a unified workflow.

Commercial grading infrastructures represent a different design paradigm. Gradescope, for instance, provides scalable rubric-based human grading, question-level organization, and LMS integration via LTI, but the evaluative process remains fundamentally human-driven rather than model-driven [48,49]. Turnitin's AI Writing Report is similarly oriented toward integrity detection and interpretability of AI-generated text proportions rather than direct pedagogical scoring, separating AI signals from similarity analysis and positioning AI as a diagnostic rather than evaluative component [50]. In contrast, the proposed framework embeds AI directly into the assessment decision through rubric-constrained reasoning grounded in retrieved course materials, producing structured, criterion-based outputs.

Finally, existing RAG-based grading systems confirm the relevance of retrieval-grounded evaluation but remain limited in integration scope. Some solutions apply RAG for rubric-aligned scoring and feedback generation in a standalone formative assessment environment [10], while another demonstrates RAG-based evaluation of open-ended responses in offline pipelines without LMS integration or structured student-facing feedback [7]. Extensions of RAG to include rubric criteria, sample answers, and historical feedback have been reported, but their implementation relies on external workflow orchestration rather than embedded LMS implementation [51]. Compared to these approaches, the proposed framework integrates RAG-grounded, rubric-constrained LLM assessment directly into Moodle, enforces structured machine-readable outputs, and extends evaluation beyond text-based tasks to multimodal UML assessment.

As summarized in Table 6, the proposed framework occupies a distinct position by combining native Moodle-connected deployment, rubric-constrained direct LLM grading, retrieval grounding in course materials, structured machine-readable result formalization, and multimodal UML assessment within one controlled orchestration architecture. Its

novelty therefore lies less in any single technological component than in the institutional integration of these components into a unified reliability-governed assessment workflow.

Table 6. Comparative positioning of the proposed framework against existing assessment systems.

System/Study	LMS Deployment	Semantic AI/LLM Grading	RAG Grounding	Machine-Readable Output	Multimodal Support	Assessment Logic
Explainable Educational Assistant integrated in Moodle [45]	Yes; native Moodle integration	Yes; semantic AI grading (BERT/CodeBERT; not generative LLM)	No/not reported	Yes; rubric-linked structured feedback and traceability	No/not reported; text and code submissions only	Semantic assessment; XAI explanation; adaptive tutoring
ChemMoodle Reacsimilarity plugin [46]	Yes; Moodle plugin	No	No	Yes; JSON/REST-based plugin output	No; structured chemical drawings assessed algorithmically	Deterministic chemistry-specific soft grading
CorrectWriting question [47]	Yes; Moodle question type/plugin	No	No	Limited; internal Moodle question output	No; text/token sequence input only	Rule-based token sequence analysis and automatic feedback
Gradescope [48,49]	External LMS/LTI connection	No; human-led grading workflow	No/not specified	Proprietary/not specified	No automated multimodal grading; scanned submissions for human grading	Human rubric-assisted grading
Turnitin AI Writing Report [50]	External LMS connection	No	No	Proprietary AI-writing report	No; prose text only	Integrity detection/instructor review
introEduAI [10]	No; standalone platform	Yes	Yes, for text-based questions; not for code-based questions	Yes; platform/API-based scoring workflow	No; text and code only	RAG-based short-answer grading; direct LLM code grading
RAG-based LLM Evaluation Pipeline [7]	No	Yes	Yes	Structured prompt output; not JSON/API payload	No	Offline RAG evaluation of open-ended responses
LLM-powered Assessment RAG System [51]	External workflow	Yes	Yes	Yes; formatted DB-stored scores and feedback	No; PDF/OCR text extraction only	External n8n-based RAG essay scoring and feedback
Proposed Moodle–n8n RAG Assessment Framework	Native Moodle-connected	Yes	Yes	Yes; JSON/rubric payload returned to Moodle	Yes; UML visual assessment	Controlled rubric-constrained grading

Note. “Not reported” indicates that the reviewed source does not explicitly describe the corresponding architectural feature. RAG grounding is marked only when retrieval of external instructional, rubric, exemplar, or reference materials is part of the grading process. Structured machine-readable output refers to outputs formatted for automated processing, API transfer, database storage, or LMS grade submission. Multimodal support is marked only when the system evaluates visual artifacts themselves, not merely when PDF or scanned files are converted to text.

The comparison highlights that existing systems typically optimize individual aspects of automated assessment. Commercial platforms such as Gradescope focus on scalable human grading workflows and rubric management, while Turnitin primarily addresses integrity monitoring and AI-content detection. In contrast, recent RAG-based grading systems demonstrate the feasibility of grounding LLM-based evaluation in course materials but remain largely standalone or externally orchestrated solutions.

However, none of the re-viewed approaches simultaneously integrate (i) rubric-constrained LLM evaluation, (ii) retrieval-grounded contextual validation, (iii) structured machine-readable output generation, (iv) multimodal assessment capability, and (v) native LMS-connected workflow orchestration within a single coherent architecture.

The proposed framework integrates these components as interdependent layers of a unified assessment pipeline. This architectural integration, rather than any single component, constitutes the primary distinguishing feature of the system.

This positioning is consistent with the empirical findings reported in Section 6, where the implemented controlled workflow demonstrates strong agreement with expert grading, structured traceability, and exploratory multimodal assessment capability. Taken together, the results and comparative analysis suggest that the primary contribution of the study lies in the design of a controlled orchestration paradigm for LLM-based assessment rather than in isolated performance optimization.

7.3. Limitations and Future Research

Despite the encouraging validation results, several limitations should be explicitly acknowledged. First, the empirical study was conducted within a relatively narrow educational context involving two parallel university course sections in Systems Engineering and a total sample of 32 students. The present findings should therefore be interpreted as initial validation evidence rather than as broad generalization across different disciplines and assessment settings.

Second, the study used a single-instructor reference grading rather than a blinded validation design with multiple raters. This choice reflects the practical reality of the course context, in which the instructor responsible for the course is also a qualified assessor using the rubric in regular teaching practice. While this provides a valid instructor-based comparison, it does not represent a broader design for generalization.

Third, the current study did not include repeated-run execution under identical prompting conditions. Consequently, execution-level reproducibility across repeated automated runs remains an important direction for future investigation.

The multilingual comparison likewise requires cautious interpretation. While the framework remained robust in both the Bulgarian and English course sections, the Bulgarian subset demonstrated somewhat stronger agreement and lower numerical deviation. Given the limited sample size, these language-related differences should be regarded as indicative rather than conclusive.

Finally, the multimodal UML analysis revealed that the present visual grading branch performs most effectively when the target criteria are structurally explicit, whereas semantically constrained and rule-dependent diagram elements remain more challenging. Future work should therefore extend validation across larger datasets, additional disciplines, repeated-run stability studies, and more fine-grained multimodal assessment scenarios.

8. Conclusions

This paper presented a role-based systematization and a controlled framework for the educational integration of LLMs in automated assessment. The study argues that the transition to fifth-generation automated assessment should not be understood only as a shift toward more powerful language models, but also as a shift toward more controlled, traceable, and pedagogically grounded assessment architectures.

The proposed framework combines six interrelated components: role prompting, rubric-constrained grading, RAG-based contextual grounding, structured machine-readable outputs, workflow orchestration, and LMS integration. Through this design, the LLM is not used as an unrestricted autonomous grader, but as one component in a controlled assessment workflow constrained by the student response, the analytical rubric, and the retrieved course-specific context.

The implemented Moodle–n8n–RAG architecture demonstrates that this framework can be operationalized in a real educational environment. The reported validation evidence

shows strong agreement with expert grading, low numerical deviation, structured traceability of results, and promising performance of the UML-based multimodal branch. These findings suggest that reliable LLM-based assessment depends not on the model alone, but on the interaction between semantic interpretation, pedagogical constraints, retrieval grounding, structured outputs, and institutional workflow integration.

The main contribution of the study is therefore methodological and architectural. It provides a reusable design pattern for integrating LLMs into automated assessment in a controlled and institutionally traceable way. This pattern is not tied to a single model or a fixed disciplinary domain, since both the rubric and the retrieved course context can be adapted to different assessment activities. However, the present validation remains limited to a software engineering course and should therefore be interpreted as initial evidence rather than as broad cross-disciplinary generalization.

Several limitations remain. The empirical evaluation was conducted with a relatively small sample in one disciplinary context; the human reference evaluation was provided by a single expert; and the study did not include repeated-run stability testing or a before/after comparison of hallucination rates with and without RAG grounding. Future work should therefore extend the validation across larger datasets, additional disciplines, multiple human raters, repeated model runs, and more diverse multimodal assessment tasks.

In conclusion, reliable educational use of LLMs in automated assessment requires a shift from model-centered evaluation to framework-centered control. The results of this study indicate that LLM-based grading becomes more methodologically defensible when it is embedded in an architecture that combines semantic capability with rubric constraints, evidential grounding, structured machine-readable outputs, and traceable LMS-based workflow integration.

Author Contributions: All authors were involved in the full process of producing this paper, including conceptualization, methodology, visualization, and preparing the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Education and Science of the Republic of Bulgaria, through the National Program D01-99: Qualification improvement in the field of nuclear technologies and nuclear engineering.

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

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

References

1. Gao, R.; Merzdorf, H.E.; Anwar, S.; Hipwell, M.C.; Srinivasa, A.R. Automatic Assessment of Text-Based Responses in Post-Secondary Education: A Systematic Review. *Comput. Educ. Artif. Intell.* **2024**, *6*, 100206. [\[CrossRef\]](#)
2. Rodrigues, L.; Xavier, C.; Costa, N.T.; Gašević, D.; Mello, R.F. Is GPT-4 Fair? An Empirical Analysis in Automatic Short Answer Grading. *Comput. Educ. Artif. Intell.* **2025**, *8*, 100428. [\[CrossRef\]](#)
3. 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. [\[CrossRef\]](#)
4. Sembey, R.; Hoda, R.; Grundy, J. Emerging Technologies in Higher Education Assessment and Feedback Practices: A Systematic Literature Review. *J. Syst. Softw.* **2024**, *211*, 111988. [\[CrossRef\]](#)
5. Sun, J.; Song, T.; Peng, W.; Song, J. A Survey of Automated Essay Scoring: Challenges, Advances, and Future. *Neurocomputing* **2025**, *650*, 130916. [\[CrossRef\]](#)
6. Tang, X.; Chen, H.; Lin, D.; Li, K. Harnessing LLMs for Multi-Dimensional Writing Assessment: Reliability and Alignment with Human Judgments. *Heliyon* **2024**, *10*, e34262. [\[CrossRef\]](#) [\[PubMed\]](#)
7. Jauhiainen, S.J.; Garagorry Guerra, A. Evaluating Students' Open-Ended Written Responses with LLMs: Using the RAG Framework for GPT-3.5, GPT-4, Claude-3, and Mistral-Large. *Adv. Artif. Intell. Mach. Learn.* **2024**, *4*, 3097–3113. [\[CrossRef\]](#)
8. Emirtekin, E. Large Language Model-Powered Automated Assessment: A Systematic Review. *Appl. Sci.* **2025**, *15*, 5683. [\[CrossRef\]](#)

9. Jacobsen, L.J.; Weber, K.E. The Promises and Pitfalls of Large Language Models as Feedback Providers: A Study of Prompt Engineering and the Quality of AI-Driven Feedback. *AI* **2025**, *6*, 35. [\[CrossRef\]](#)
10. Mendonça, P.C.; Quintal, F.; Mendonça, F. Evaluating LLMs for Automated Scoring in Formative Assessments. *Appl. Sci.* **2025**, *15*, 2787. [\[CrossRef\]](#)
11. Nkoyo, T.A.F.E.; Ijezue, C.F.; Amjad, A.I.; Amjad, M.; Butt, S.; Castañeda-Garza, G. Advances in Auto-Grading with Large Language Models: A Cross-Disciplinary Survey. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, Vienna, Austria, 31 July–1 August 2025; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 477–498. [\[CrossRef\]](#)
12. García-Varela, F.; Nussbaum, M.; Mendoza, M.; Martínez-Troncoso, C.; Bekerman, Z. ChatGPT as a Stable and Fair Tool for Automated Essay Scoring. *Educ. Sci.* **2025**, *15*, 946. [\[CrossRef\]](#)
13. Grévisse, C. LLM-Based Automatic Short Answer Grading in Undergraduate Medical Education. *BMC Med. Educ.* **2024**, *24*, 1060. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Seßler, K.; Fürstenberg, M.; Bühler, B.; Kasneci, E. Can AI Grade Your Essays? A Comparative Analysis of Large Language Models and Teacher Ratings in Multidimensional Essay Scoring. In *Proceedings of the 15th International Learning Analytics and Knowledge Conference*, Dublin, Ireland, 24–28 March 2025; pp. 462–472. [\[CrossRef\]](#)
15. Paranjape, B.; Lundberg, S.; Singh, S.; Hajishirzi, H.; Zettlemoyer, L.; Ribeiro, M.T. ART: Automatic Multi-Step Reasoning and Tool-Use for Large Language Models. *arXiv* **2023**, arXiv:2303.09014. [\[CrossRef\]](#)
16. Singh, J.; Magazine, R.; Pandya, Y.; Nambi, A.U. Agentic Reasoning and Tool Integration for LLMs via Reinforcement Learning. *arXiv* **2025**, arXiv:2505.01441.
17. Vangelova, A.; Gancheva, V. AI-Based Automated Scoring Layer Using Large Language Models and Semantic Analysis. *Appl. Sci.* **2026**, *16*, 3537. [\[CrossRef\]](#)
18. Sychev, O.; Anikin, A.; Prokudin, A. Automatic Grading and Hinting in Open-Ended Text Questions. *Cogn. Syst. Res.* **2020**, *59*, 264–272. [\[CrossRef\]](#)
19. Shermis, M.D.; Burstein, J.; Apel Bursky, S. Introduction to Automated Essay Evaluation. In *Handbook of Automated Essay Evaluation: Current Applications and New Directions*; Shermis, M.D., Burstein, J., Eds.; Routledge/Taylor & Francis Group: New York, NY, USA, 2013; pp. 1–15.
20. Dikli, S. An Overview of Automated Scoring of Essays. *J. Technol. Learn. Assess.* **2006**, *5*, 1–36. Available online: <https://ejournals.bc.edu/index.php/jtla/article/view/1640> (accessed on 18 December 2025).
21. Mizumoto, A.; Eguchi, M. Exploring the Potential of Using an AI Language Model for Automated Essay Scoring. *Res. Methods Appl. Linguist.* **2023**, *2*, 100050. [\[CrossRef\]](#)
22. Attali, Y.; Burstein, J. Automated Essay Scoring with e-rater® V.2. *J. Technol. Learn. Assess.* **2006**, *4*, 3.
23. Birla, N.; Kumar Jain, M.; Panwar, A. Automated Assessment of Subjective Assignments: A Hybrid Approach. *Expert Syst. Appl.* **2022**, *203*, 117315. [\[CrossRef\]](#)
24. Pecuchova, J.; Benko, L.; Drlik, M. Automated Grading of Open-Ended Questions in Higher Education Using GenAI Models. *Int. J. Artif. Intell. Educ.* **2025**, *35*, 3813–3846. [\[CrossRef\]](#)
25. 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. [\[CrossRef\]](#)
26. Cipriano, E.; Ferrato, A.; Limongelli, C.; Schicchi, D.; Taibi, D. Leveraging Large Language Models to Assist Teachers in Code Grading. In *Artificial Intelligence in Education*; Cristea, A.I., Walker, E., Lu, Y., Santos, O.C., Isotani, S., Eds.; Springer Nature: Cham, Switzerland, 2025; Volume 15880, pp. 204–217. [\[CrossRef\]](#)
27. Bernik, A.; Radošević, D.; Čep, A. A Comparative Study of Large Language Models in Programming Education: Accuracy, Efficiency, and Feedback in Student Assignment Grading. *Appl. Sci.* **2025**, *15*, 10055. [\[CrossRef\]](#)
28. Papachristou, I.; Dimitroulakos, G.; Vassilakis, C. Automated Test Generation and Marking Using LLMs. *Electronics* **2025**, *14*, 2835. [\[CrossRef\]](#)
29. Ndukwe, I.G.; Amadi, C.E.; Nkomo, L.M.; Daniel, B.K. Automatic Grading System Using Sentence-BERT Network. In *Artificial Intelligence in Education*; Bittencourt, I.I., Cukurova, M., Muldner, K., Luckin, R., Millán, E., Eds.; Springer International Publishing: Cham, Switzerland, 2020; Volume 12164, pp. 224–227. [\[CrossRef\]](#)
30. Dada, I.D.; Akinwale, A.T.; Tunde-Adeleke, T.-J. A Structured Dataset for Automated Grading: From Raw Data to Processed Dataset. *Data* **2025**, *10*, 87. [\[CrossRef\]](#)
31. Balakrishnan, R.M.; Pati, P.B.; Singh, R.P.; S, S.; Kumar, P. Fine-Tuned T5 for Auto-Grading of Quadratic Equation Problems. *Procedia Comput. Sci.* **2024**, *235*, 2178–2186. [\[CrossRef\]](#)
32. Li, J.; Gui, L.; Zhou, Y.; West, D.; Aloisi, C.; He, Y. Distilling ChatGPT for Explainable Automated Student Answer Assessment. *arXiv* **2023**, arXiv:2305.12962. [\[CrossRef\]](#)
33. Pack, A.; Barrett, A.; Escalante, J. Large Language Models and Automated Essay Scoring of English Language Learner Writing: Insights into Validity and Reliability. *Comput. Educ. Artif. Intell.* **2024**, *6*, 100234. [\[CrossRef\]](#)

34. Lee, G.-G.; Latif, E.; Wu, X.; Liu, N.; Zhai, X. Applying Large Language Models and Chain-of-Thought for Automatic Scoring. *Comput. Educ. Artif. Intell.* **2024**, *6*, 100213. [\[CrossRef\]](#)
35. Organisciak, P.; Acar, S.; Dumas, D.; Berthiaume, K. Beyond Semantic Distance: Automated Scoring of Divergent Thinking Greatly Improves with Large Language Models. *Think. Ski. Creat.* **2023**, *49*, 101356. [\[CrossRef\]](#)
36. Chu, S.; Kim, J.; Wong, B.; Yi, M. Rationale Behind Essay Scores: Enhancing S-LLM's Multi-Trait Essay Scoring with Rationale Generated by LLMs. *arXiv* **2025**, arXiv:2410.14202. [\[CrossRef\]](#)
37. Seneviratne, H.M.T.W.; Manathunga, S.S. Artificial Intelligence Assisted Automated Short Answer Question Scoring Tool Shows High Correlation with Human Examiner Markings. *BMC Med. Educ.* **2025**, *25*, 1146. [\[CrossRef\]](#) [\[PubMed\]](#)
38. Oğuz, E. Can Generative AI Figure Out Figurative Language? The Influence of Idioms on Essay Scoring by ChatGPT, Gemini, and Deepseek. *Assess. Writ.* **2025**, *66*, 100981. [\[CrossRef\]](#)
39. Xu, W.; Kassim, M.S.S.; Hoo, W.L.; Yang, W.; Xu, T. Explainable AI for Education: Enhancing Essay Scoring via Rubric-Aligned Chain-of-Thought Prompting. *Int. J. Mod. Phys. C* **2026**, *37*, 2542013. [\[CrossRef\]](#)
40. Kinder, A.; Briese, F.J.; Jacobs, M.; Dern, N.; Glodny, N.; Jacobs, S.; Leßmann, S. Effects of Adaptive Feedback Generated by a Large Language Model: A Case Study in Teacher Education. *Comput. Educ. Artif. Intell.* **2025**, *8*, 100349. [\[CrossRef\]](#)
41. Seo, H.; Hwang, T.; Jung, J.; Kang, H.; Namgoong, H.; Lee, Y.; Jung, S. Large Language Models as Evaluators in Education: Verification of Feedback Consistency and Accuracy. *Appl. Sci.* **2025**, *15*, 671. [\[CrossRef\]](#)
42. Landis, J.R.; Koch, G.G. The Measurement of Observer Agreement for Categorical Data. *Biometrics* **1977**, *33*, 159–174. [\[CrossRef\]](#)
43. Taghipour, K.; Ng, H.T. A Neural Approach to Automated Essay Scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2016; pp. 1882–1891. [\[CrossRef\]](#)
44. Ludwig, S.; Mayer, C.; Hansen, C.; Eilers, K.; Brandt, S. Automated Essay Scoring Using Transformer Models. *Psych* **2021**, *3*, 897–915. [\[CrossRef\]](#)
45. Villegas-Ch, W.; Gutierrez, R.; García-Ortiz, J.; Guevara, V. Explainable Educational Assistant Integrated in Moodle: Automated Semantic Assessment and Adaptive Tutoring Based on NLP and XAI. *Discov. Artif. Intell.* **2025**, *5*, 191. [\[CrossRef\]](#)
46. Plyer, L.; Marcou, G.; Perves, C.; Bonachera, F.; Varnek, A. Implementation of a Soft Grading System for Chemistry in a Moodle Plugin: Reaction Handling. *J. Cheminform.* **2024**, *16*, 90. [\[CrossRef\]](#) [\[PubMed\]](#)
47. Sychev, O. Questions for Teaching Phrase Building with Automatic Feedback. *Softw. Impacts* **2023**, *15*, 100461. [\[CrossRef\]](#)
48. Gradescope Guides. Using Gradescope LTI 1.3 with Moodle as an Instructor. Available online: <https://guides.gradescope.com/hc/en-us/articles/23587719150349-Using-Gradescope-LTI-1-3-with-Moodle-as-an-Instructor> (accessed on 24 April 2026).
49. Gradescope Guides. Grading Submissions with Rubrics. Available online: <https://guides.gradescope.com/hc/en-us/articles/22249389005709-Grading-submissions-with-rubrics> (accessed on 24 April 2026).
50. Turnitin Guides. Using the AI Writing Report. Available online: <https://guides.turnitin.com/hc/en-us/articles/22774058814093-Using-the-AI-Writing-Report> (accessed on 24 April 2026).
51. Barenji, R.V.; Salimi, N.; Khoshgoftar, S. An LLM-Powered Assessment Retrieval-Augmented Generation (RAG) for Higher Education. *arXiv* **2026**, arXiv:2601.06141. [\[CrossRef\]](#)

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.