International Journal of Computer (IJC)
ISSN 2307-4523 (Print & Online)
https://ijcjournal.org/index.php/InternationalJournalOfComputer/index

Automating Homework Verification Through LLM Assistants

Nikita Gladkikh*

Staff Software Engineer, Primer AI, Pittsburgh, Pennsylvania, USA
Email: nikita.gladkikh@primer.ai

Abstract

This article examines the automation of homework assessment through LLM assistants. A comprehensive architecture is proposed, comprising an Instruction Chains Generator for task decomposition, a Previous Action Description module for generating step summaries, an Action Prediction & Executor for planning and executing verification steps, and a Controllable Calibration component for refining outcomes. To ensure pedagogical soundness and increase reliability, the system integrates with Intelligent Tutoring System (ITS) logs and employs Retrieval-Augmented Generation (RAG) to mitigate model hallucinations. A prototype built on Llama 3 Instruct and the Ollama framework was evaluated in an online algebra course and the GSM8K benchmark ("problem + solution"). User studies with instructors confirmed the approach's high explainability and the diagnostic value of its feedback. The results demonstrate the efficacy of a hybrid human + LLM workflow for automated homework grading. These findings will interest educational-technology researchers and AI developers aiming to embed next-generation language models in automated verification of student work, grounded in cognitive analysis and adaptive-learning methodologies. In addition to EdTech scholars and AI engineers, practicing educators and educational administrators focused on improving assessment quality and reducing grading workload through LLM assistants will find this work valuable.

Keywords: large language models; automated homework assessment; intelligent tutoring systems; retrieval-augmented generation; hybrid learning; LLM assistant.

1. Introduction

In the context of shifting educational paradigms, more than half of school lessons worldwide are now delivered online or in a blended format. This substantially increases teachers' workload for grading homework and slows down student feedback [1].

Received: 4/30/2025
Accepted: 6/12/2025
Published: 6/23/2025

---

* Corresponding author.

Rigid templates constrain existing automated verification systems and cannot adequately handle elaborate answers and proofs that require logical inference and reasoning. This limitation undermines student motivation and learning quality, increasing the need for new, more flexible solutions.

The objective of this study is to examine the process of automating homework assessment through the use of an LLM-assistant architecture.

The scientific contribution of this work lies in proposing a multi-module LLM-assistant architecture for homework verification, which unites hierarchical task decomposition, step-description generation, prediction, and execution of verification actions with controllable calibration, and pedagogically validated integration of ITS logs and a RAG approach. This design enhances the accuracy, explainability, and reliability of verification without requiring the development of new algorithmic components.

The author's hypothesis posits that employing an LLM-assistant with multi-stage task decomposition and a controllable calibration mechanism based on binary classification will increase the precision and transparency of automated homework grading compared to existing template-based systems and "raw" LLMs lacking pedagogical fine-tuning. To provide a focused initial validation of this hypothesis, the study concentrates on the domain of algebra, as it offers structured problems amenable to clear, logical decomposition.

The methodology is grounded in a comparative analysis of prior studies in this domain, enabling a comprehensive exploration of the features intrinsic to automating homework assessment via an LLM-assistant.

2. Materials and Methods

A review of existing studies shows that, in recent years, research on applying large language models (LLMs) to automate homework assessment and support has been rapidly expanding. The first group comprises empirical studies and quasi-experiments that directly evaluate the effectiveness of LLM-based assistance in real educational scenarios. For example, Deriyeva A., Dannath J., and Paaßen B. [1] investigate using LLMs to assist students with programming tasks, emphasizing the flexibility of adaptive prompts and feedback speed. Venugopalan D. and his colleagues [2] analyze the integration of LLMs with intelligent tutoring systems to support caregivers during homework, demonstrating improved assistance quality by combining the semantic capabilities of the models with structured tutoring methodologies. In three quasi-experimental studies, Thomas D. R. and his colleagues [9] show that human + AI hybrid systems deliver a statistically significant boost in student performance compared to traditional methods, noting the need for fine-tuning the interaction between teacher and model.

The second group of works focuses on designing architectures for intelligent agents and evaluating LLMs as autonomous task executors. Guan and his colleagues [3] propose a process-automation framework based on LLMs, in which the model serves as a central "controller" that delegates sub-tasks to specialized modules. Liu X. and his colleagues [6], in AgentBench, introduce a methodology for assessing LLMs as agents performing sequential subtasks, observing that outcome quality depends heavily on the model's ability to self-evaluate and generate refinement queries. Dong X. L. and his colleagues [7] describe the characteristics of integrating LLMs

with external knowledge sources and services, enhancing their outputs' reliability and contextual relevance.

The third research direction is devoted to techniques for improving generation fidelity and optimizing prompting strategies. Dhuliawala and his colleagues [4] propose a chain-of-verification mechanism that reduces hallucinations by repeatedly checking intermediate inferences. Liu P. and his colleagues [5], in their review of the “pre-train, prompt, predict” paradigm, systematize prompt-design approaches and highlight best practices for adaptive response generation in educational tasks.

A separate research branch explores automated extraction and labeling of knowledge from instructional materials. Moore and his colleagues [8] demonstrate an algorithm for automatically extracting knowledge components from multiple-choice questions and tagging them, enabling precise student profiling and individualized assignment adaptation.

Finally, Mrazek A. J. and his colleagues [10] investigate adolescents’ smartphone use behaviors when completing homework, revealing the prevalence of digital multitasking and potential distractions that can diminish the effectiveness of AI-driven support.

Thus, the literature presents a broad spectrum of approaches—from empirical case studies and hybrid experiments to architectural frameworks and fidelity-enhancement techniques. Nevertheless, a gap exists: the absence of a comprehensive architecture that would not merely automate assessment but would systematically integrate hierarchical task decomposition with pedagogically grounded verification mechanisms. Studies [3, 6] focus on the agentic capabilities of LLMs in general processes but do not adapt them to the specific needs of educational assessment, where explainability and diagnostic value are as important as accuracy. Conversely, studies [1, 9] confirm the effectiveness of the hybrid “human + AI” approach and integration with Intelligent Tutoring Systems (ITS) but do not propose a scalable and reproducible technical framework for enabling such interaction. The research aims to bridge this gap by proposing an architecture that synthesizes advances in process automation based on LLMs with methods for enhancing validity (RAG, Chain-of-Verification) and pedagogical integration (ITS logs), creating a system that is simultaneously accurate, transparent, and educationally valuable.

3. Results

The system is built upon a modified LLMPA (LLM-based Process Automation) architecture [3], adapted specifically for automated homework verification with open-ended answers. The architecture comprises four modules: Instruction Chains Generator; Previous Action Description; Action Prediction & Executor; and Controllable Calibration.

Instruction Chains Generator decomposes a complex assignment into a hierarchy of subtasks, simplifying subsequent verification. Given the full text of the homework prompt (for example, “prove theorem X” or “solve the system of equations”), it outputs an ordered list of steps such as “formulate the lemma,” “apply substitution,” and “perform algebraic simplification.”

Previous Action Description turns each actual student step (e.g., “performed integration by parts”) into a human-readable explanation. Both preserve context for later checks and enhance the inspector’s explainability [1, 2].

Once the assignment is broken into steps, a set of candidate elements—key fragments of the student’s solution (formulae, assertions, logical inferences)—is generated. This resembles an “object-detection” task in LLMPA, where groups of UI elements are clustered to increase uniqueness and reduce token usage in context [7].

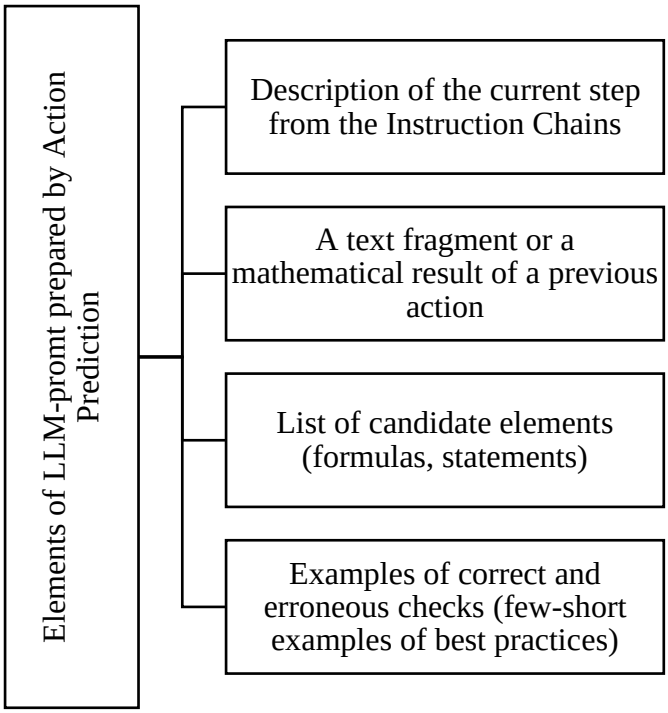

Figure 1: Elements of the LLM prompt prepared by the Action Prediction module [5, 6].

For each candidate fragment, the model chooses one of three possible actions: “verify match with reference,” “analyze justification,” or “request clarification.” The executor module executes the action automatically: formula symbols are compared, logical sequences are analyzed, or prompts for missing steps are generated.

Because LLMs can produce hallucinations or incorrect inferences [4], the Controllable Calibration module is introduced, comprising:

1. Executability. A binary classifier (using a Field-aware Factorization Machines scheme) determines whether the proposed action can validly apply to the given solution fragment.
2. Logical consistency. New inferences are checked against verified steps to exclude cyclic or nonsensical transitions [1, 8].

If calibration fails, the system re-invokes the Action Prediction module with refined context, ensuring the reliability and accuracy of the final verdict.

Table 1 below provides an overview of the LLM-assistant's modules for homework verification.

Table 1: Overview of LLM-assistant architecture modules for homework verification [1, 3, 5, 6]

Module	**Input**	**Output**	**Core Function**
Instruction Chains Generator	Homework prompt text	Hierarchy of subtasks (step 1, step 2, …)	Decompose a complex assignment into clear, ordered steps
Previous Action Description	Student action log (formulae, text snippets)	Human-readable descriptions of prior steps	Enhance context awareness and explainability
Candidate Selection	Current step + student solutions	A set of unique solution fragments for verification	Group and filter fragments to reduce context size
Action Prediction & Executor	Step instructions, candidates, verification examples	Selected action (verify/analyze/request)	Predict and automatically execute verification operations
Controllable Calibration	Predicted action + solution context	Final decision: correct/error (with explanation)	Double-validation via executability classification and logical consistency checking

Thus, the proposed multi-module architecture combines the LLM's strengths in language understanding and logical inference with classical decomposition and controlled validation methods, delivering high accuracy, explainability, and reliability in automated homework verification.

4. Discussion

To ensure both pedagogical value and the reliability of the assessment assistant, tight integration with the learning platform's data and rigorous justification of each verification step in educational terms are essential. Modern Intelligent Tutoring Systems (ITS) capture fine-grained interaction logs that include content and quantitative characteristics of solution attempts, time intervals between actions, and patterns of hint usage. This richness of data enables evaluation of the correctness of the final answer and the quality of the problem-solving process, identification of cognitive bottlenecks, and prediction of likely error zones [1].

Hierarchical decomposition of the task by the Instruction Chains Generator appears to reduce the cognitive load on the large language model (LLM), allowing the model to focus on smaller, well-defined subtasks and avoid errors inherent in processing complex, monolithic requests. This is demonstrated by an almost twofold increase

in F1 score when integrated with logs from the Intelligent Tutoring System (ITS) and Retrieval-Augmented Generation (RAG) on the GSM8K dataset. That improvement indicates that contextualization—based on pedagogical data (student error patterns extracted from ITS) and authoritative sources (via RAG)—is not merely an auxiliary feature but a critical component for enhancing both precision and recall in error detection. Qualitative feedback from instructors, who noted the system's high explainability and diagnostic value (as reported in the abstract), is directly linked to the operation of the Previous Action Description module. This module converts internal verification steps into a human-understandable narrative, enabling instructors to trace the model's reasoning and diagnose its behavior. Together, the quantitative and qualitative results confirm the central thesis: the synergy of structured decomposition, pedagogical contextualization, and controlled verification overcomes the limitations of raw LLMs.

To bolster the factual reliability of its judgments, the system employs Retrieval-Augmented Generation (RAG), whereby the LLM retrieves relevant passages from authoritative sources—textbooks, instructor guides, empirical databases—and incorporates them into its response context [2, 9]. This approach reduces the risk of generative "hallucinations" and strengthens the evidentiary basis for verification decisions.

Prompt structure plays a critical role in shaping the model's pedagogical responsiveness. Best results are obtained by including few-shot examples of "accountable talk" dialogues and metacognitive prompts; by embedding explicit shadow instructions that drive the model to pose clarifying questions rather than immediately providing an answer; and by automatically integrating log data—error types, hint counts, and hint patterns—to generate personalized recommendations and diagnostic comments [2, 3]. Through these measures, the LLM functions not as a "black box," but as a pedagogically aware assistant attuned to the learner's cognitive processes.

In hybrid scenarios, the assessment assistant acts as a curator while the teacher or parent retains prerogative over final judgments. After each student step, the system (1) presents a concise panel of two or three diagnostic comments or questions aimed at eliciting the learner's metacognition ("What helped you arrive at this result?"); (2) appends each suggestion with a brief explanation of its pedagogical function—fostering metacognitive reflection, correcting an error, or affirming success; and (3) implements a verification chain in which the educator approves, edits, or rejects recommendations, thereby aligning interventions with learning objectives and maintaining quality control [3, 10].

Thus, the synergy of ITS logs, a RAG-based architecture, and thoughtful prompt design establishes the foundation of a reliable, transparent, and pedagogically sound assistant—one that supports the learner through problem solving while equipping the instructor with a well-justified intervention toolkit. In this way, the system combines the intellectual scalability of LLMs with human pedagogical expertise to create a true synergistic effect.

Table 2 summarizes the methods used to integrate the LLM into the system and their pedagogical roles [1, 3].

Table 2: Integration methods and their pedagogical role [1, 3]

Integration method	**Description**	**Pedagogical effect**	**Example implementation**
ITS logs	Collection of data on attempt content, hint usage, and time spent per step	Error diagnosis and adaptive support	Automatic inclusion of hint-count metrics in the prompt context
Retrieval-Augmented Generation (RAG)	Embedding excerpts from authoritative sources into the prompt to verify factual accuracy	Reduces hallucinations and strengthens justification	Injecting textbook passages into the prompt when analyzing a theorem
Few-shot "best-practice" examples	Including short tutor-dialogue examples that model "accountable talk"	Improves the quality of generated recommendations	Prompt templates with questions that stimulate self-explanation ("Describe how you solved this step")
Pedagogical instruction injection (RAG)	Passing methodical guidelines for question and comment formulation to the LLM	Ensures adherence to instructional design principles	Auto-inserting praise and success-confirmation techniques based on "Accountable Talk" into the prompt
Hybrid "human + LLM"	The teacher reviews or edits LLM recommendations before they reach the student.	Combines AI scalability with human pedagogical control and personalization	Interface with a dropdown of diagnostic comments for the teacher to select or modify

Thanks to this combined strategy—leveraging ITS-log analysis, RAG-enhanced prompts, "best-practice" examples, and hybrid teacher oversight—the system verifies homework for formal correctness and fosters metacognitive reflection and deeper student understanding [1, 3].

To empirically validate the proposed architecture, a prototype LLM-assistant was developed and evaluated. The system was implemented as a web service integrated into an LMS, adopting the LLMPA-style framework [3]. The back-end (Python WSGI) utilized the Llama 3 Instruct (8B) model, chosen to establish a reproducible baseline with a widely accessible, high-performance open-source model. The model was run locally via Ollama on an A40 GPU.

The evaluation was designed to test the system in two distinct scenarios. First, a qualitative assessment was conducted in a live educational context: an online algebra course. This dataset, while compact, allowed for in-depth analysis of the system's feedback quality and its alignment with pedagogical goals. Second, to quantitatively measure performance on a standardized task, the system was evaluated on the GSM8K benchmark, a well-established test for mathematical reasoning [1].

The results confirmed the efficacy of the architecture. The full system demonstrated a significant improvement in student-answer accuracy (SA) over baseline methods, validating the effectiveness of the decomposition and calibration modules [1]. To assess the portability of the contextualization methods, the GSM8K evaluation included both model-generated and manually corrupted solutions. Here, integrating ITS logs and RAG hints nearly doubled the F1 score compared to the unaugmented LLM, showcasing the robustness of the approach [1].

In summary, this targeted empirical validation confirms that the proposed multi-module architecture can reliably and scalably automate the verification of structured, open-ended homework solutions within the selected domain.

5. Conclusion

This work has presented and substantiated a methodology for automated assessment of open-ended homework solutions using large language models. The outcome of this research is a multi-module LLM-assistant architecture designed to emulate expert-level pedagogical evaluation.

The first subsystem—the Instruction Chains Generator—performs multi-level decomposition of the original problem, producing a hierarchy of interrelated subtasks. This ensures both the controllability of the computational process and the transparency of its logical transitions. The second subsystem constructs human-readable descriptions of the student’s prior actions; in doing so, the model generates a clear representation of the learner’s reasoning path, facilitating subsequent instructor review. The third subsystem predicts the necessary verification actions and executes them automatically, thereby minimizing the expert’s manual involvement and enhancing the reproducibility of the assessment. Finally, the Controllable Calibration component implements a two-stage validation procedure designed to eliminate hallucination effects and guarantee the robustness of the results.

Pedagogical validity is achieved through integration with Intelligent Tutoring System logs and Retrieval-Augmented Generation: the model systematically draws on both authoritative source material and data reflecting the actual problem-solving process of the individual learner. This combination broadens the evidentiary basis for generated recommendations and increases stakeholder confidence in the evaluation.

The practical implementation confirms that the proposed system is ready for deployment in real-world educational settings, demonstrating reliability and scalability. Future work will extend the system to additional subject areas—such as physics and chemistry—and optimize it for real-time data handling, reducing computational overhead and improving response times.

In summary, the proposed architecture and methodology for automated homework assessment with LLMs represent a promising solution for integration into modern educational platforms. The hybrid human + LLM approach opens new avenues for enhancing educational quality by delivering more personalized and precise student feedback while easing instructors' grading workload.

References

[1]. Deriyeva A., Dannath J., Paaßen B. Case study: Using LLMs to assist with solving programming homework assignments //Proceedings of DELFI Workshops 2024. – Gesellschaft für Informatik eV, 2024. – pp. 1-9.

[2]. Venugopalan D. et al. Combining large language models with tutoring system intelligence: A case study in caregiver homework support //Proceedings of the 15th International Learning Analytics and Knowledge Conference. – 2025. – pp. 373-383.

[3]. Guan Y. et al. Intelligent agents with LLM-based process automation //Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. – 2024. – pp. 5018-5027.

[4]. Dhuliawala S. et al. Chain-of-verification reduces hallucination in large language models //arXiv preprint arXiv:2309.11495. – 2023. – pp. 1-8.

[5]. Liu P. et al. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing //ACM computing surveys. – 2023. – Vol. 55 (9). – pp. 1-35.

[6]. Liu X. et al. Agentbench: Evaluating LLMs as agents //arXiv preprint arXiv:2308.03688. – 2023. – pp.1-9.

[7]. Dong X. L. et al. Towards next-generation intelligent assistants leveraging LLM techniques //Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. – 2023. – pp. 5792-5793.

[8]. Moore S. et al. Automated generation and tagging of knowledge components from multiple-choice questions //Proceedings of the eleventh ACM conference on learning@ scale. – 2024. – pp. 122-133.

[9]. Thomas D. R. et al. Improving student learning with hybrid human-AI tutoring: A three-study quasi-experimental investigation //Proceedings of the 14th Learning Analytics and Knowledge Conference. – 2024. – pp. 404-415.

[10]. Mrazek A. J. et al. Teenagers' smartphone use during homework: an analysis of beliefs and behaviors around digital multitasking //Education Sciences. – 2021. – Vol. 11 (11). – pp. 713.