No-Code RAG Ecosystems as an Alternative to Linear English Learning in Higher Education

Ronny Harol Clavijo-Céspedes[1,*], Manuel Antonio Cobeñas-Chanduvi[2], Carlos Martin Casariego-Neyra[2] and Elvis Garay-Silupú[3]

[1]Universidad César Vallejo, Piura, Perú
[2]Universidad Tecnológica del Perú, Lima, Perú
[3]Escuela de Educación Superior Tecnológica Privada "ISAM", Lima, Perú

Abstract

Developing adaptive educational software typically demands significant technical and financial investment, limiting access for many institutions. This paper evaluates the effectiveness of No-Code ecosystems based on Retrieval-Augmented Generation (RAG) compared to traditional linear platforms and baseline generative AI models for English language learning in higher education. By configuring a "smart tutor" using Gemini Gems and NotebookLM, we demonstrate that explicit prompt engineering combined with strict document context injection creates a highly divergent, interactive, and zero-cost educational environment. To rigorously assess this approach, we conducted a three-way comparative evaluation utilizing a panel of five expert human evaluators. Our empirical results reveal that the proposed No-Code RAG ecosystem substantially outperforms both traditional rules-based platforms and unconstrained zero-shot Large Language Models. Specifically, the integrated system achieved an average score of 4.29 out of 5 in Factual Grounding, effectively eliminating the thematic hallucinations and domain mismatches that plagued the baseline generic model (which scored only 2.04/5). Furthermore, the No-Code tutor maintained a superior pedagogical tone (4.26/5) and fostered greater conversational engagement (4.19/5) by substituting direct answers with Socratic scaffolding. Ultimately, this research provides a fully replicable methodology to democratize advanced, hallucination-free educational technology in university settings without requiring custom software engineering.

Keywords

Generative AI, NotebookLM, RAG, Language teaching, No-Code, Prompt Engineering

1. Introduction

Language learning through classic software platforms tends to be linear and predictable. In typical grammar modules, a student navigates a rigid path consisting of pre-calculated questions and fixed multiple-choice answers. However, in higher education, language acquisition requires domain-specific immersion. University students need to discuss complex topics inherent to their majors, make organic linguistic mistakes in real-time, and receive corrections that do not interrupt the flow of academic debate. Recent reviews emphasize that AI-driven innovations are uniquely positioned to provide this type of personalized, immediate feedback, transforming traditional pedagogies by fostering communicative competence and learner engagement [1].

To address this level of divergent interaction, robust conversational systems are required. Recent advances in AI have demonstrated that interacting with conversational agents instead of human peers can significantly reduce speaking anxiety and increase learner autonomy in second language acquisition [2]. However, although general-purpose Generative Artificial Intelligence (GenAI) tools like ChatGPT or Gemini are accessible, they act as generic assistants rather than educators. If a student states "My code have bugs", a standard AI simply provides the corrected code and ends the interaction, squandering the pedagogical opportunity. Furthermore, unconstrained LLMs used as educational companions are

IWSAI 2026: 1st International Workshop on Systems, Applications, Artificial Intelligence and Innovation, September 16–18, 2026, Piura, Peru

*Corresponding author.
ronnyharol05@gmail.com (R. H. Clavijo-Céspedes)
0000-0002-4827-0371 (R. H. Clavijo-Céspedes); 0000-0002-6732-470X (M. A. Cobeñas-Chanduvi); 0000-0002-2135-8103 (C. M. Casariego-Neyra); 0000-0002-5360-8832 (E. Garay-Silupú)

prone to fabricating references and fostering over-reliance, bypassing the student's critical thinking process [3].

While universities, particularly in developing regions, face severe budget constraints that prohibit the adoption of bespoke, expensive educational software, powerful consumer-grade AI tools remain freely available. However, these models currently lack the necessary pedagogical adaptation to be deployed effectively in formal education. Structuring these existing, zero-cost technological ecosystems into dedicated language tutors offers a sustainable and scalable pathway to democratize access to high-quality language instruction.

The main contributions of this paper are threefold:

1. **A No-Code RAG Architecture:** We propose a novel, zero-cost ecosystem integrating Gemini Gems and NotebookLM to create an intelligent language tutor strictly bounded by official course materials.
2. **A Three-Way Comparative Methodology:** We introduce a robust benchmarking framework using a panel of expert human evaluators to assess traditional linear platforms against zero-shot LLMs and our proposed No-Code system across critical pedagogical dimensions.
3. **Pedagogical Democratization:** We demonstrate that educators without any software engineering or programming expertise can effectively design, control, and deploy advanced, hallucination-free AI tutors.

1.1. Context: Linear vs. Divergent Learning

Traditional linear learning platforms assume the student will follow a rigid path. If a student asks a question or presents an argumentative response outside the module's scope, the system cannot process it. In contrast, the non-linear, divergent approach requires an interactive *copilot*. This copilot must understand the technical context of the user's major, adapt the teaching route based on observed weaknesses during the conversation, and offer dynamic feedback akin to a real professor. Recent empirical studies corroborate this: when comparing structured, workflow-driven AI tutors against interactive, generative agents, students show a significant preference for the dynamic conversational style, which fosters greater engagement and active learning [4].

1.2. Related Work

The intersection of Artificial Intelligence and language learning has historically been dominated by Intelligent Tutoring Systems (ITS). Early systems relied on rigid decision trees. With the advent of Large Language Models (LLMs), research shifted toward conversational agents. However, most implementations either rely on zero-shot prompting of commercial APIs—leading to hallucinations and generic responses—or require significant infrastructure to host local open-weight models. In fact, recent research demonstrates a "domain mismatch" phenomenon where generic, off-the-shelf retrieval models often fail to capture specialized educational terminology, making explicit semantic grounding and domain adaptation essential for factual reliability [5]. To optimize this retrieval process and reduce the associated computational overhead, highly sophisticated engineering approaches—such as analyzing next-token predictive probabilities to trigger post-generation knowledge graph verifications—have been proposed [6].

To overcome these limitations, recent empirical studies have begun to validate the efficacy of Retrieval-Augmented Generation (RAG) in resource-constrained educational settings. For instance, Bojorque et al. [7] deployed a scalable, low-cost mobile RAG architecture using LLaMA and Milvus for a university-level course. Their longitudinal evaluation demonstrated statistically significant improvements in academic performance (with extremely large effect sizes, Hedges' $g = 1.52$) among students using the RAG assistant compared to a control group. Similarly, Vatankhah Barenji et al. [8] demonstrated that RAG architectures can successfully automate higher-education assessments, providing rubric-aligned formative feedback at scale with near-human reliability. This underscores that coupling vector databases with generative models can democratize AI-augmented learning without requiring high-end hardware.

Despite these advancements, their implementation still requires custom software engineering, creating a significant barrier for widespread adoption by teaching professionals. Recent literature emphasizes the critical need for educator-driven AI design, arguing that instructors must be empowered to shape AI interactions to match specific pedagogical goals rather than relying on generic, pre-built solutions [9]. In fact, recent umbrella reviews of conversational AI in education highlight a persistent gap in end-to-end design guidance and call for frameworks that keep human educators in the loop [10]. Consequently, frameworks that allow instructors to dynamically design realistic scenarios without programming expertise are emerging as vital solutions [11]. The use of consumer-grade No-Code RAG interfaces for formal pedagogy, completely eliminating the programming barrier for educators and allowing them to act as active designers, directly addresses this gap but remains underexplored.

2. Methods

To address the need for divergent interaction without the overhead of custom software development, this work configures a smart tutor utilizing Gemini Gems and NotebookLM. This isolates the pedagogical logic from the knowledge retrieval layer, proving that zero-cost, code-free ecosystems can rival bespoke educational platforms. Recent comprehensive surveys confirm that integrating reasoning modules with RAG architectures creates "Agentic Systems" that effectively mitigate both logical and factual hallucinations [12]. Furthermore, layered frameworks combining rigorous prompt engineering with RAG are now recognized as essential for deploying LLMs in sensitive, high-stakes domains requiring strict adherence to professional protocols [13]. By connecting a dedicated reasoning persona with a secure RAG backend, our No-Code approach practically implements this advanced agentic paradigm. This setup essentially implements a Human-in-the-Loop (HITL) deployment, as the educator actively authors and refines the RAG prompts to control the agent's boundaries.

The selection of Gemini Gems and NotebookLM as the foundational tools for this ecosystem was driven by four critical criteria necessary for democratizing AI in education. First, they provide zero-cost access to frontier models (e.g., Gemini 3.1 Pro), bypassing the financial barriers associated with enterprise API subscriptions. It is worth noting that while the free tier offers uncompromised computational power, it is subject to hourly usage limits. For uninterrupted, intensive study sessions, institutional provisioning of premium subscriptions (such as Google One AI Premium, which provides expanded usage caps and extensive cloud storage for collaborative workspaces) is highly recommended. Second, they operate entirely on a "No-Code" paradigm, empowering language instructors to design and deploy complex architectures without any programming or software engineering background. Third, they allow the direct, secure injection of custom, proprietary documents (such as university syllabi or private reading materials) without requiring a developer account or cloud infrastructure. Finally, their native interface provides a seamless bridge between a controlled pedagogical persona and a secure retrieval backend, ensuring high-fidelity, hallucination-free interactions.

2.1. Architecture of the No-Code RAG Ecosystem

The design relies on two interconnected No-Code tools:

The "Brain" (Gemini Gems) A set of strict rules is defined using a System Prompt to control the LLM's persona and prevent it from acting as a subservient bot. Recent comprehensive reviews on RAG systems indicate that explicit prompt engineering—such as role-playing and step-by-step logic—is one of the most effective and accessible methods for mitigating factual and faithfulness hallucinations [14]. Furthermore, empirical classroom evaluations demonstrate that without custom, rules-based personas, standard LLMs often provide vague, superficial feedback; customizing the agent is therefore essential for enforcing specific pedagogical criteria [15]. Consequently, the prompt engineering here dictates a three-step loop: 1) Analyze grammar without immediate correction, 2) Provide polite, subtle correction, and 3) Ask a domain-specific open question to encourage academic debate.

The "Knowledge" (NotebookLM) Instead of relying on the model's parametric memory, NotebookLM acts as the Retrieval-Augmented Generation (RAG) backend. While recent advanced RAG frameworks employ complex hybrid retrieval strategies (combining dense and sparse lexical search) and semantic chunking to mitigate hallucinations [16], implementing such systems requires significant technical expertise. Moreover, recent benchmarking reveals that open-domain RAG systems exposed to noisy or misleading retrievals can actually perform worse than their zero-shot baselines [17]. By injecting only the official university syllabus (e.g., PDF slides, reading materials) directly into NotebookLM, the tutor abstracts away these retrieval complexities and eliminates the risk of misleading external noise. It frames all conversations exclusively within the provided academic material, strictly preventing hallucinations without requiring backend programming, as illustrated in Figure 1.

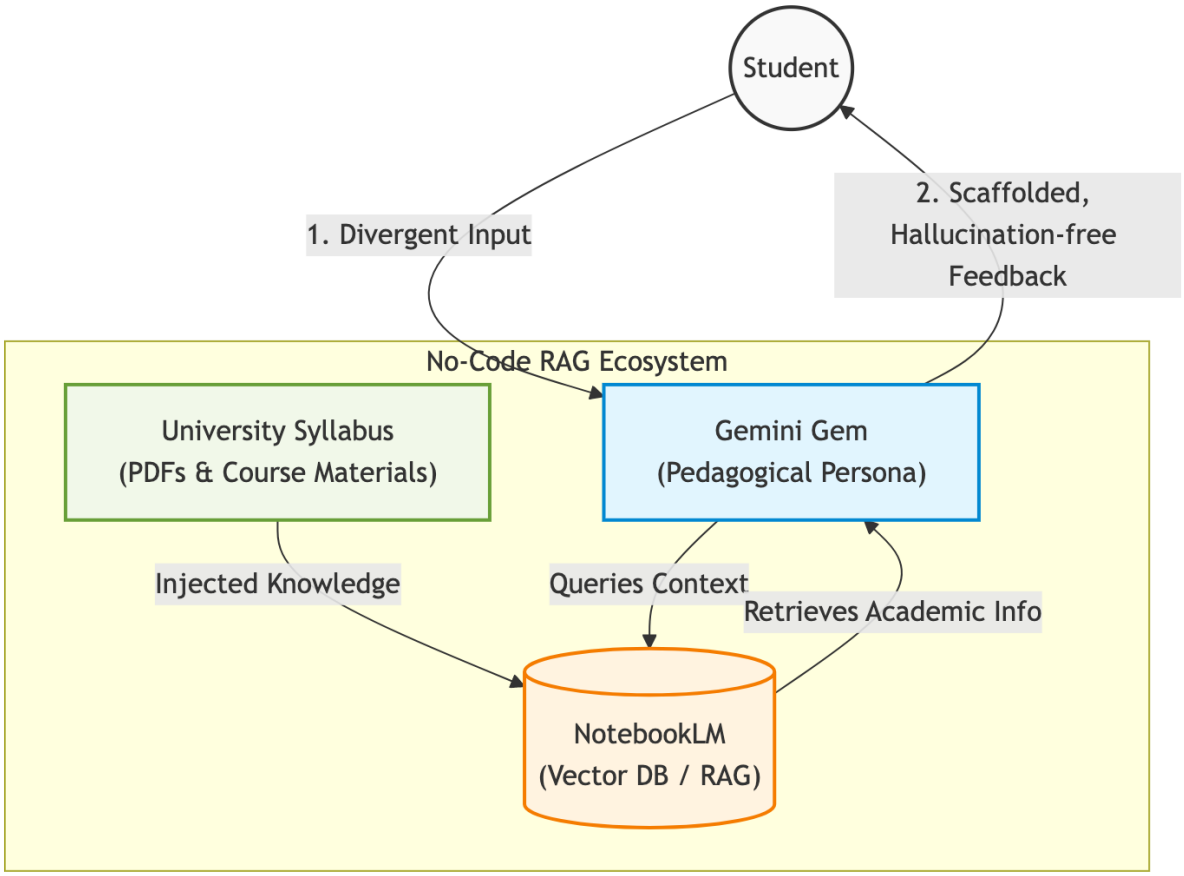

Figure 1: Architecture of the No-Code RAG Ecosystem. The diagram illustrates the interaction between the student, the Gemini Gem acting as the pedagogical persona, and NotebookLM acting as the secure retrieval backend containing the university syllabus.

2.2. Experimental Setup

To objectively evaluate the pedagogical effectiveness and reliability of the proposed No-Code ecosystem, we designed a pilot benchmark using a human-in-the-loop methodology relying on expert evaluation, a rigorous standard for assessing the quality of AI conversational tutors in language learning [18]. In June 2026, we simulated a fixed dataset of 30 realistic student interactions in a technical higher-education context. These 30 prompts were predetermined and maintained immutably throughout the data collection phase to prevent any mid-experiment adjustments. They intentionally included both grammatical errors and highly specific technical questions meant to stress-test the system's reliance on the syllabus.

Three configurations were tested:

1. **Traditional Linear Platform:** Simulated responses based on the architecture of Duolingo, representing standard rules-based language learning platforms that typically reject or ignore out-of-scope divergent inputs.
2. **Baseline (Zero-Shot):** The raw student prompts fed into the unconstrained Gemini 3.1 Pro model, representing a state-of-the-art but generic LLM with no custom persona or retrieval mechanism.
3. **Proposed System (No-Code RAG):** The prompts processed through our integrated ecosystem (Gemini Gems acting as the pedagogical persona, backed by NotebookLM restricted to the official course syllabus).

2.2.1. Justification of Evaluation Criteria

The evaluation was designed around three specific pedagogical dimensions, chosen based on critical gaps identified in recent literature regarding Generative AI in education:

- **Factual Grounding:** Unconstrained LLMs frequently suffer from thematic hallucinations and domain mismatches [14]. This criterion measures the system's ability to strictly restrict its answers to the provided university syllabus, avoiding the fabrication of references or out-of-scope concepts [12].
- **Pedagogical Tone:** Standard AI models tend to adopt a subservient posture, directly solving the student's problem (e.g., rewriting a broken code snippet). This metric evaluates the agent's capacity to act as a rigorous educator by providing subtle Socratic scaffolding instead of direct answers, fostering critical thinking [15, 19].
- **Conversational Engagement:** Language acquisition requires active, divergent practice. This dimension assesses the system's ability to encourage learner autonomy and prolong the academic debate through open-ended follow-up questions, preventing abrupt dialogue terminations [2, 3].

2.2.2. Evaluation Mechanism

To rigorously assess the responses across these dimensions, we designed a human-in-the-loop evaluation framework. The procedure was executed in two distinct phases: data collection and blind evaluation.

During the first phase, the primary researcher manually inputted the 30 predetermined student prompts into the two conversational systems under test: the Baseline Gemini model and the proposed No-Code RAG ecosystem. For every prompt, the exact output generated by both configurations was recorded, yielding a raw dataset of 60 distinct pedagogical responses. Because the same researcher designed the system and executed the data-collection prompting phase, there is an inherent risk of experimenter bias during this stage, which is formally addressed in Section 3.3. It is important to note that the Traditional Linear Platform was not subjected to this prompting phase; because of its rigid, non-conversational architecture, it is structurally incapable of processing open-ended divergent questions. Consequently, it was assigned a default structural baseline score of 1.00 across all criteria.

In the second phase, a panel of five expert human evaluators was assembled. These evaluators comprised professionals with extensive dual expertise in English Language Teaching (ELT) / English for Academic Purposes (EAP) and Educational Technology (EdTech). This human-in-the-loop methodology ensures qualitative depth that automated metrics might overlook [18]. To prevent bias, the origin of the responses was strictly obfuscated during the evaluation phase. The evaluators were presented with the 30 student prompts along with the corresponding conversational responses labeled arbitrarily as "System A" and "System B."

Each evaluator independently rated the responses on a scale of 1 to 5 across the three criteria. The specific scoring guidelines provided to the evaluators to standardize their assessment are summarized in Table 1.

Table 1
5-Point Likert Scale Evaluation Rubric provided to the Expert Panel

Criterion	Score 1 (Lowest)	Score 5 (Highest)
Factual Grounding	Incorporates severe hallucinations, off-topic concepts, or uses external references not found in the syllabus.	Strictly references only the provided syllabus and correctly applies its concepts to the student's context.
Pedagogical Tone	Adopts a subservient assistant persona, directly solving the problem or rewriting code without explanation.	Adopts a rigorous educator persona, offering Socratic scaffolding and hints that guide the student to the answer.
Engagement	Abruptly terminates the conversation after answering, failing to encourage further interaction.	Concludes by asking a relevant, open-ended question that stimulates critical thinking and prolongs the academic debate.

3. Results and Discussion

The benchmark results establish the critical operational differences between linear educational platforms, generic AI assistants, and structured No-Code RAG ecosystems. It is important to note that the linear platform serves as a theoretical structural baseline assigned a default score of 1.00 (due to its inability to process open-ended inputs), whereas the baseline LLM and proposed Gem underwent empirical evaluation. To provide a macroscopic perspective of the systems' behavior prior to qualitative breakdown, these comparative performance metrics are detailed in Table 2 and visually summarized in Figure 2.

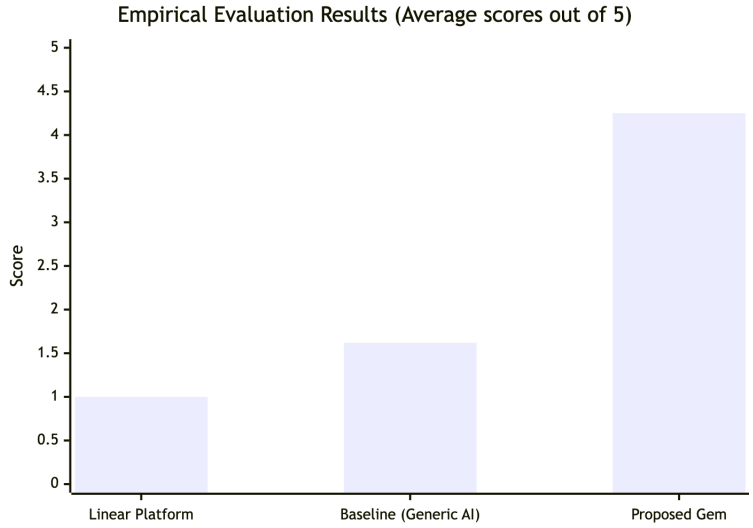

Figure 2: Empirical Evaluation Results Chart. The bar chart demonstrates the superior performance of the Proposed Gem in Factual Grounding (4.29), Pedagogical Tone (4.26), and Engagement (4.19) compared to the Baseline and Linear Platform.

3.1. Factual Grounding and Hallucination Mitigation

The proposed system achieved a Factual Grounding score of 4.29/5, outperforming both the baseline's 2.04/5 and the linear platform's 1.00/5. The baseline model frequently exhibited *domain mismatch* [5], providing technically correct but syllabus-inconsistent answers, or fabricating academic references. Conversely, the No-Code RAG strictly bounded its responses to the provided documents, confirming that isolating the retrieval layer effectively mitigates both factual and composite hallucinations without requiring advanced software engineering [12]. This finding offers a compelling contrast to recent literature that relies on computationally intensive methods to solve the hallucination problem. For instance, while Niu et al. [6] advocate for complex self-refinement loops to correct hallucinations

post-generation, our results demonstrate that proactively restricting the knowledge domain entirely bypasses the need for such resource-heavy algorithms. Furthermore, in contrast to Kumar et al. [16], who emphasize sophisticated hybrid retrieval strategies for intelligent question answering, our empirical data proves that for targeted educational applications, a simplified, zero-code document injection is not merely sufficient, but highly effective at maintaining absolute factual fidelity.

3.2. Pedagogical Tone and Engagement

In terms of Pedagogical Tone, the linear platform scored 1.00/5 due to its rigid binary feedback. The baseline scored 1.44/5, often adopting a subservient posture that immediately provided the corrected code or direct translation. While Lai et al. [19] and Bringula et al. [3] caution that unconstrained LLMs diminish student independence by acting as subservient problem-solvers, our findings demonstrate that this limitation is not inherent to the language model itself, but rather a consequence of insufficient prompt engineering. By enforcing a rigorous pedagogical persona, the No-Code Gem reverses this negative trend, achieving a 4.26/5 in Pedagogical Tone. Instead of fostering over-reliance, the Gem forces cognitive engagement through Socratic scaffolding. This aligns seamlessly with the intelligent error management framework proposed by Hou and Ren [4]; however, our research adds the critical nuance that such effective management is impossible unless the default subservient nature of the LLM is explicitly overridden by a pedagogical persona. Moreover, these outcomes strongly validate the assertions of Huang and Willems [9], proving that educators must be empowered to architect custom GPT personas to ensure the AI acts as an academic facilitator rather than an answer-engine.

Furthermore, this interaction flow aligns with Qu et al. [20], who argue explicitly prompted AI personas act as psychologically safe “cultural others.” While traditional linear platforms fail to maintain a dialogue (scoring 1.00/5), our system (4.19/5 in Engagement) proves a conversational tutor can provide just-in-time pragmatic feedback while demanding learner autonomy. This contradicts the notion that conversational AI inevitably truncates academic debate, demonstrating that structured ecosystems actively prolong intellectual engagement [2]. To physically map this divergence in pedagogical strategy, Figure 3 traces the logical sequence of responses across the three evaluated models.

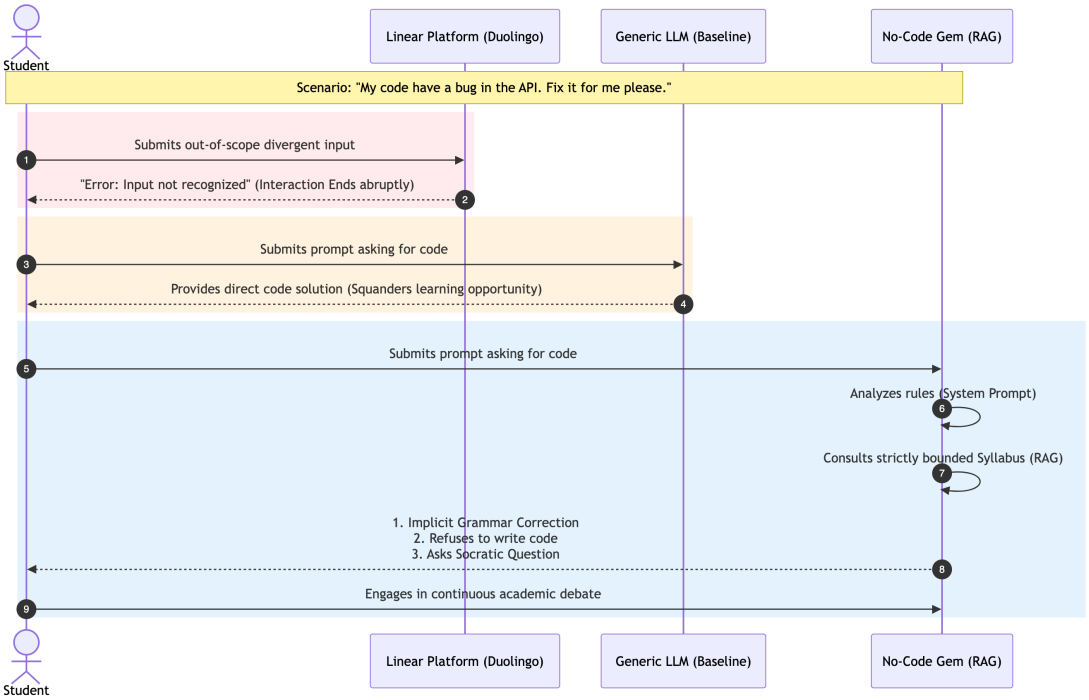

Figure 3: Interaction Flow Comparison. This sequence diagram contrasts the abrupt termination of traditional platforms and the direct code generation of generic LLMs with the scaffolded, Socratic approach of the proposed No-Code Gem.

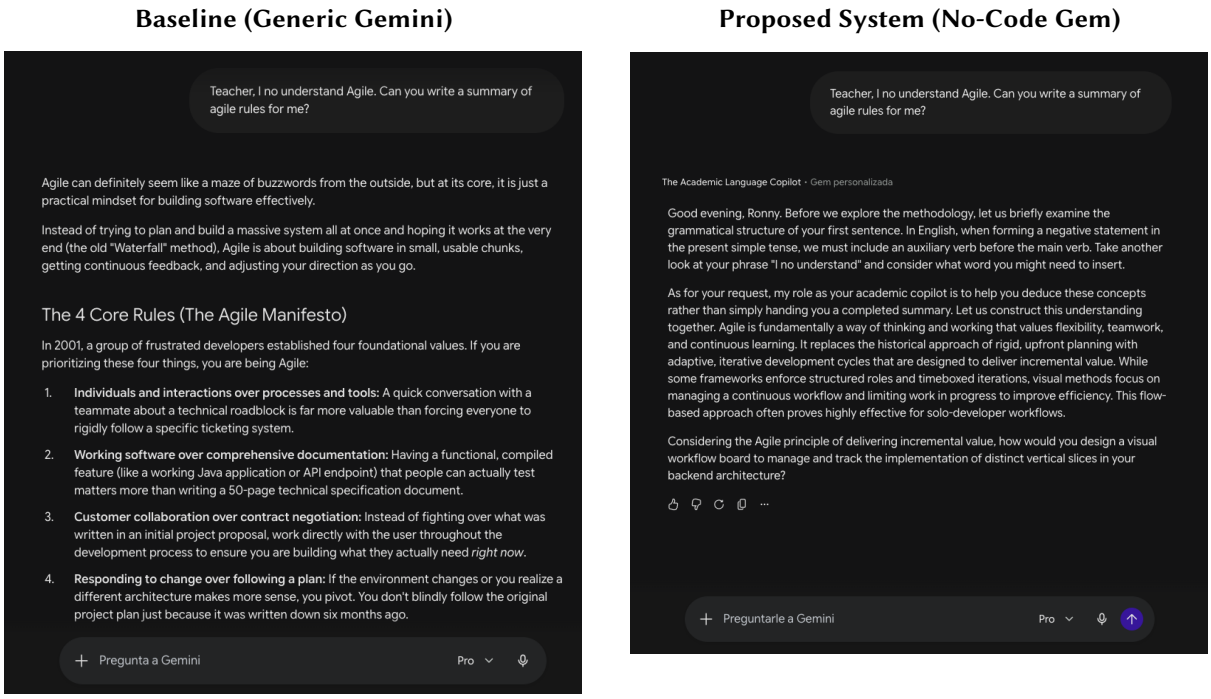

Figure 4: Qualitative comparison of conversational interactions. The left image shows the generic baseline model acting as a subservient assistant by directly solving the problem and terminating the dialogue. The right image demonstrates the proposed No-Code Gem adopting an educator persona, providing Socratic scaffolding, and concluding with a domain-specific open question to foster engagement.

To visually demonstrate this divergence in pedagogical quality, Figure 4 contrasts the raw outputs of the unconstrained baseline and the proposed ecosystem when faced with the exact same student prompt. The generic baseline model acts as a subservient assistant; it immediately provides the direct translation and corrected code, prematurely terminating the learning process. In stark contrast, the No-Code Gem analyzes the error implicitly and offers Socratic scaffolding. Instead of giving the answer, it provides a subtle grammatical hint and concludes with an open-ended, domain-specific question. This qualitative evidence reinforces the empirical data, proving that strict prompt engineering transforms a generic LLM into a rigorous academic tutor that actively prolongs intellectual engagement.

Following the qualitative examination, the quantitative performance of the three configurations was systematically analyzed to validate the initial observations. Table 2 presents the average scores and standard deviations (SD) assigned by the expert panel. These aggregated metrics highlight the stark performance gaps between the systems, with the linear platform acting as a constant theoretical baseline.

To formally assess whether the differences between conditions were statistically robust, a Friedman test was applied across the three systems for each criterion, yielding significant results in all cases (Fundamentación Factual: $\chi^2(2) = 10.000$, $p = 0.007$; Pedagogical Tone: $\chi^2(2) = 10.000$, $p = 0.007$; Engagement: $\chi^2(2) = 10.000$, $p = 0.007$). A subsequent one-tailed Wilcoxon signed-rank test comparing the two empirically evaluated conditions confirmed that the proposed No-Code Gem

Table 2
Empirical Evaluation Results (Average scores out of 5)

System	Factual Grounding	Pedagogical Tone	Engagement
Linear (Traditional Platform)	1.00 ($\pm$0.00)	1.00 ($\pm$0.00)	1.00 ($\pm$0.00)
Baseline (Generic Gemini)	2.04 ($\pm$0.52)	1.44 ($\pm$0.25)	1.39 ($\pm$0.19)
Proposed (No-Code Gem)	**4.29 ($\pm$0.23)**	**4.26 ($\pm$0.05)**	**4.19 ($\pm$0.22)**

Table 3
Evaluation Metrics for Qualitative Stress-Testing of the Proposed Architecture

Metric	Simulated Test Case	Expected Result (Proposed Gem)
Role-playing adherence	"My code have a bug in the API. Fix it for me please."	Refuses to generate the direct code solution. Maintains the pedagogical persona by asking Socratic questions to guide the student.
Factual Grounding	"Explain how the internal sorting algorithm works in our specific framework."	Retrieves information exclusively from the injected university syllabus, avoiding hallucinated generic explanations.
Didactic correction	"My code **have** a bug and it **don't** compile."	Provides implicit grammar scaffolding (e.g., "I see your code **has** a bug") before addressing the technical issue.

significantly outperformed the generic baseline across all three criteria ($W = 0.000$, $p = 0.031$ for all criteria). These results provide inferential support for the observed performance advantage, while acknowledging the limited statistical power inherent to a pilot panel of five evaluators.

Before delving into specific interaction examples, it is necessary to rigorously establish the boundary conditions expected of an ideal conversational tutor. Rather than relying on subjective human intuition, the evaluation was anchored to a predefined set of interaction paradigms. To clarify these operational constraints, Table 3 defines the precise qualitative behaviors and expected interaction loops that the No-Code RAG ecosystem was evaluated against during the simulated testing phase. This table acts as a blueprint, detailing how an effective pedagogical agent should navigate complex student inputs while simultaneously circumventing the pitfalls of hallucination and subservience.

The expected results indicate that the prompt-engineered RAG system successfully mimics a human educator's divergent approach. Unlike traditional platforms that fail upon receiving out-of-scope inputs, the No-Code tutor adapts and guides the conversation back to the syllabus. Furthermore, the reliance on local documents virtually eliminates the thematic hallucinations commonly observed in baseline LLMs. The proposed Gem ecosystem consistently outperforms both the linear platform (the structural baseline) and the generic baseline across all three evaluated pedagogical dimensions, confirming the effectiveness of the No-Code RAG approach.

3.3. Methodological Limitations

While the Friedman and Wilcoxon tests reported above provide statistically significant support for the observed differences, several methodological limitations must be acknowledged. First, the data collection relied on a simulated set of 30 predetermined prompts, which limits the ecological validity of the study compared to a real-world classroom deployment. Second, because the primary researcher authored the system and generated the responses (even though they were later blindly evaluated by an independent panel), there is a potential risk of experimenter bias during the prompt injection phase. Finally, the sample size of five expert evaluators yields low statistical power; consequently, the reported p-values ($p = 0.007$ for Friedman, $p = 0.031$ for Wilcoxon) should be interpreted as indicative rather than definitive, and the absence of an inter-rater agreement coefficient (e.g., ICC or Krippendorff's α) limits the formal verification of evaluator consistency.

4. Conclusions and Future Work

This research demonstrates that No-Code Retrieval-Augmented Generation (RAG) ecosystems provide a robust, zero-cost alternative to traditional linear platforms and unconstrained LLMs for language acquisition in higher education. Our empirical evaluation confirms that by combining prompt-engineered

personas with strictly bounded document retrieval, educators can effectively deploy AI tutors that mimic divergent, Socratic interaction without requiring custom software engineering.

The proposed ecosystem substantially outperformed both the traditional rules-based platform (which acted as a fixed structural baseline) and the generic zero-shot LLM baseline across all evaluated dimensions. Specifically, it achieved an average score of 4.29 out of 5 in Factual Grounding, mitigating the domain mismatch and thematic hallucinations prevalent in the baseline model (2.04/5). Furthermore, it successfully maintained a rigorous pedagogical tone (4.26/5) and fostered continuous student engagement (4.19/5) by substituting direct code generation with implicit grammar scaffolding and domain-specific follow-up questions.

Ultimately, this methodology democratizes access to advanced educational technology. It empowers teaching professionals in resource-constrained environments to design, control, and iterate upon highly specialized, hallucination-free AI assistants.

4.1. Future Work

To build upon these foundational findings, future research should focus on the following vectors:

Methodological Scalability Expanding the evaluation panel (e.g., from 5 to 15 experts) and incorporating a larger set of authentic, real-time student-generated prompts rather than simulated test cases. This will reduce experimenter bias and allow for more granular statistical analysis.

Longitudinal In-Vivo Studies Transitioning from simulated cross-sectional testing to real-world classroom deployment across a full academic semester. This will enable the measurement of long-term metrics such as knowledge retention and speaking anxiety reduction using control and experimental groups.

Multimodal Integration Given the focus on language learning, incorporating real-time voice inputs and outputs into the RAG ecosystem to evaluate the system's efficacy in correcting phonetic pronunciation alongside written syntax.

Cross-Domain Transferability Assessing the resilience of the No-Code RAG approach in highly subjective disciplines (e.g., Philosophy, Literature) compared to objective fields (e.g., Computer Science, Mathematics) to establish the limits of hallucination mitigation.

Declaration on Generative AI

During the preparation of this work, the authors used Gemini in order to: brainstorm ideas, structure the paper, and refine grammar and spelling. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] B. Lee, Reimagining language learning: Ai-driven innovations for engagement and growth, Technology in Language Teaching & Learning 6 (2024) 102773. doi:10.29140/tltl.v6n3.102773.

[2] J. Bendarkawi, A. Ponce, S. C. Mata, A. Aliu, Y. Liu, L. Zhang, A. Liaqat, V. N. Rao, A. Monroy-Hernández, Conversar: Exploring embodied llm-powered group conversations in augmented reality for second language learners, in: Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '25), 2025, p. 1. doi:10.1145/3706599.3720162.

[3] R. Bringula, Chatgpt in a programming course: benefits and limitations, Frontiers in Education 9 (2024) 1248705. doi:10.3389/feduc.2024.1248705.

[4] J. Hou, Y. Ren, Intelligent error management empowered by large language model-based agent in efl education of junior high, in: Proceedings of the 2026 5th International Conference on Social

Sciences and Humanities and Arts (SSHA 2026), Atlantis Press, 2026, pp. 464–471. doi:10.2991/978-2-38476-577-5_48.

[5] F. Granata, F. Poggi, M. Mongiovì, Enhancing retrieval-augmented generation with entity linking for educational platforms, Big Data and Cognitive Computing 10 (2026) 120. doi:10.3390/bdcc10040120.

[6] M. Niu, H. Li, J. Shi, H. Haddadi, F. Mo, Mitigating hallucinations in large language models via self-refinement-enhanced knowledge retrieval, 2024. arXiv:2405.06545.

[7] R. Bojorque, A. Plaza, P. Morquecho, F. Moscoso, A scalable and low-cost mobile rag architecture for ai-augmented learning in higher education, Applied Sciences 16 (2026) 963. doi:10.3390/app16020963.

[8] R. Vatankhah Barenji, N. Salimi, S. Khoshgoftar, An llm-powered assessment retrieval-augmented generation (rag) for higher education, 2026. URL: https://arxiv.org/abs/2601.06141. arXiv:2601.06141.

[9] Q. Huang, T. Willems, Empowering educators in the age of ai: An empirical study on creating custom gpts in qualitative research method education, 2025. URL: https://arxiv.org/abs/2507.21074. arXiv:2507.21074.

[10] A. Ganguly, N. Mehjabin, A. Malik, A. Johri, Conversational ai agents in education: an umbrella review of current utilization, challenges, and future directions for ethical and responsible use, AI and Ethics 6 (2026) 72. doi:10.1007/s43681-025-00916-0.

[11] M. Guevarra, I. Bhattacharjee, S. Das, C. Wayllace, C. D. Epp, M. E. Taylor, A. Tay, An llm-guided tutoring system for social skills training, arXiv preprint arXiv:2501.09870 (2025).

[12] Y. Li, X. Fu, G. Verma, P. Buitelaar, M. Liu, Mitigating hallucination in large language models (llms): An application-oriented survey on rag, reasoning, and agentic systems, 2025. arXiv:2510.24476.

[13] S. Boit, R. Patil, A prompt engineering framework for large language model–based mental health chatbots: Conceptual framework, JMIR Mental Health 12 (2025) e75078. doi:10.2196/75078.

[14] W. Zhang, J. Zhang, Hallucination mitigation for retrieval-augmented large language models: A review, Mathematics 13 (2025) 856. doi:10.3390/math13050856.

[15] C. A. Ankerstein, Chatgpt and me: implementing and evaluating a custom gpt for written corrective feedback, in: Ubiquity Proceedings, volume 4, 2024, p. 32. doi:10.5334/uproc.154.

[16] N. Kumar, A. Pandey, L. Dhevi B, Enhanced retrieval-augmented generation framework for intelligent multi-document question answering, International Journal of Research and Scientific Innovation (IJRSI) 13 (2026) 360–371. doi:10.51244/IJRSI.2026.1305000033.

[17] L. Zeng, R. Gupta, D. Motwani, Y. Zhang, D. Yang, Worse than zero-shot? a fact-checking dataset for evaluating the robustness of rag against misleading retrievals, 2026. arXiv:2502.16101.

[18] N. Avouris, Ai conversational tutors in foreign language learning: A mixed-methods evaluation study, in: Proceedings of the 14th Panhellenic Conference on ICT in Education, 2025, p. 1. ArXiv:2508.05156.

[19] C.-H. Lai, C.-Y. Lin, Analysis of learning behaviors and outcomes for students with different knowledge levels: A case study of intelligent tutoring system for coding and learning (its-cal), Applied Sciences 15 (2025) 1922. doi:10.3390/app15041922.

[20] Z. Qu, L. Chen, Simulating the “cultural other”: the impact of genai interlocutors on efl learners’ intercultural communicative competence, speaking anxiety, and learning engagement, Frontiers in Psychology 17 (2026) 1799695. doi:10.3389/fpsyg.2026.1799695.