

Collaborative Feedback Generation in Intelligent Tutoring Systems through Collective Intelligence

Thyago Tenório¹, Seiji Isotani²

¹Campus Penedo – Federal University of Alagoas
S/N Beira Rio Avenue – 57200-000 – Penedo – AL – Brazil

²Graduate School of Education – University of Pennsylvania
3700 Walnut Street – PA 19104 – Philadelphia – USA

thyago.oliveira@penedo.ufal.br, sisotani@upenn.edu

Abstract. *Inner loops play a critical role in intelligent tutoring systems (ITS) by providing the step-by-step feedback necessary to guide students through tasks and enhance their learning experience. However, delivering this detailed feedback requires highly granular domain knowledge, placing a substantial burden on domain experts. While automatic approaches are typically limited to specific contexts, both manual and automatic methods often ignore the potential of student interactions within the system. To address this challenge, it was proposed a novel, hybrid approach to designing the knowledge model with inner loops by leveraging students' collective intelligence through data derived from their interactions. The approach was evaluated within an ITS with 147 students, and the expert evaluations showed high accuracy, precision, and F-scores. Consequently, our approach rapidly generates highly accurate ITS inner loops while requiring significantly less expert involvement than traditional methods. To the best of our knowledge, this is the first domain-independent approach that integrates collective intelligence into authoring of ITS inner loops.*

1. Introduction

Intelligent Tutoring Systems (ITS) are advanced educational technologies that leverage artificial intelligence to optimize learning outcomes. These systems have the capability to represent knowledge [Polson and Richardson 2013] and offer meaningful, detailed feedback to students as they engage in complex problem-solving tasks [Psootka et al. 1988]. By emulating the pedagogical strategies of human tutors, ITS facilitate highly personalized learning experiences tailored to each student's needs [Nkambou et al. 2010], yielding learning gains comparable to those achieved through one-on-one human tutoring [VanLehn 2011]. A fundamental mechanism enabling this personalization is the inner loop, which provides fine-grained, step-by-step feedback throughout the student's problem-solving process [Vanlehn 2006]. This contrasts sharply with traditional tutoring systems, which typically deliver coarse-grained feedback based solely on final answers [VanLehn 2011].

Although inner loops facilitate step-by-step problem-solving [Vanlehn 2006] and yield significant learning gains [VanLehn 2011], most Intelligent Tutoring Systems (ITS) lack them entirely or implement them in a highly simplified manner [Vanlehn 2006]. This occurs because to deliver meaningful, high-level feedback, the inner loops requires a more

extensive and high-granularity domain knowledge, which is a very challenging task for its designer and for the domain experts. First, because it is necessary to have all domain knowledge supporting information related to step-by-step resolution [Vanlehn 2006] and, second, an effective process is required to instantiate such detailed domain knowledge. The current models are inadequate, dependent on a specific context and/or the system interface, and the instantiation of information is heavily dependent of human experts.

In an attempt to bring the students to a more active role in the process of designing the domain knowledge and reduce specialist overload, a novel hybrid approach was proposed to designing the domain knowledge with inner loops of intelligent tutoring systems by leveraging students' collective intelligence through data derived from their interactions. Accordingly, this article introduces an context and interface independent step-by-step resolution model and it proposes the use of Collective Intelligence (CI) of the students in the process of authoring the domain knowledge, co-producing the required information between the domain experts and the data of the student interactions in the system or directly from the students in a continuous process managed by the system, which has the direct benefits of reducing the teacher's overload while empowering students as active participants in the learning ecosystem.

CI has demonstrated significant efficacy across various domains and holds high potential for educational applications [Meza et al. 2018]. Even though CI is not a novel concept, its combination with online educational technologies is an emerging area that brings positive impacts in the educational field [Tenório et al. 2022]. The application of CI in the process of domain knowledge authoring has recently started [Tenório et al. 2020, Tenório et al. 2023], and consequently, the results of this approach remain underexplored. The inclusion of the active participation of students in this process could reduce the cost of running the ITS [Sun et al. 2016, Zaidan and Callison-Burch 2011], minimize the time to create the information [Cross et al. 2014], and ensure the scalable development of tutoring models [Vaish et al. 2018].

However, there are concerns about the quality of information generated by students, compared to that produced by domain experts [Tenório et al. 2022]. Although preliminary studies suggest that the results produced by students can achieve equivalent quality to the results produced by experts [Tenório et al. 2022] [Tenório et al. 2021], there is no definitive evidence in the literature that the authoring domain knowledge using CI will be effective. In this sense, it becomes necessary to investigate whether authoring domain knowledge using the collective intelligence of the students presents equivalent quality when compared to creation using specialists. Thus, the main contributions of this work are threefold: (1) a novel model for ITS inner loops driven by collective intelligence; (2) an active, student-engaged instantiation process for authoring domain knowledge; and (3) an empirical validation of the proposed framework, including a expert evaluation of the generated outputs. The remainder of this paper is structured as follows. Section 2 describes background concepts and related work. Section 3 details the proposed model and authoring process. Section 4 presents the results and discussion, and Section 5 concludes with recommendations for future research.

2. Background and Related Works

Implementing inner loops requires extensive, fine-grained domain knowledge, which poses a formidable challenge for both system designers and domain experts [VanLehn 2011, Aleven et al. 2009, Tacoma et al. 2021]. However, given that this authoring process is a complex and onerous task for the domain expert, many ITS do not have inner loops entirely, while others implement them in highly simplified manner that restricts the provided feedback. For instance, the physics tutor described in [Vanlehn 2006] evaluates only the student’s definitive answer; such single-loop systems capture only a single piece of evidence per task, inevitably limiting their pedagogical effectiveness [Vanlehn 2006].

On the other hand, notable examples of ITS with inner loops include the Algebra Cognitive Tutor [Pane et al. 2014], Andes [Schulze et al. 2000] and the SQL-Tutor [Mitrovic 2003]. However, the domain models driving these systems are inherently constrained to specific contexts, and their authoring relies on domain experts. While some automatic approaches attempt to mitigate this dependency by utilizing AI techniques for domain modeling [MacLellan et al. 2014, Weitekamp et al. 2020, Seffrin et al. 2012], their applications remain largely domain-specific. Authoring tools designed to facilitate the creation of ITS with domain-independent inner loops, such as CTAT [Aleven et al. 2006] and SimStudent [MacLellan et al. 2014], still suffer from prolonged authoring times due to expert overload and the difficulty of constructing comprehensive models. Consequently, there remains a critical gap in the literature regarding effective methods for representing inner-loop domain models, as existing approaches are typically context and interface dependent, and exclusively reliant on experts for content authoring.

This exclusive reliance on experts creates a severe bottleneck that delays system deployment, compromises content scalability, and can ultimately degrade the quality of the tutoring experience. Conversely, student interactions generate a wealth of data during the step-by-step resolution process, which information could significantly assist other students but is typically discarded by the system. Integrating students into the ITS domain knowledge authoring process has the potential to substantially expand system capabilities [Heffernan et al. 2016] while benefiting both educators and learners. Within this context, it is proposed leveraging the concept of Collective Intelligence (CI) to enhance the authoring process. CI is broadly defined as groups of individuals acting collectively in ways that seem intelligent [Leimeister 2010]. By applying this framework, the fine-grained information required for inner loops is co-produced by students and teachers through a continuous, system-managed process.

A prominent example of an ITS adopting a related collaborative concept is ASSISTments [Heffernan and Heffernan 2014], a platform that crowdsources educational content — such as tips, explanations, immediate feedback [Razzaq et al. 2020], common error responses, videos, and new problems — by leveraging a vast network of teachers. This approach facilitates content creation across multiple contexts and significantly reduces individual teacher workload. However, the authoring model implemented in ASSISTments operates at a coarse-grained (task) level and still relies entirely on experts. This distinction highlights the novelty of our proposal, which focuses on fine-grained inner loops and directly harnesses the collective intelligence of students.

3. Authoring Domain Knowledge via Collective Intelligence

To provide fine-grained, step-by-step feedback, an ITS must contain an integrated representation of all possible resolution paths, mapping every potential state, step, and action within a unified knowledge network. Accordingly, this knowledge was formalized as a knowledge graph, defined as $(\mathbb{KG} = (V, A))$, where the vertex set V corresponds to all possible problem states ($\mathbb{E}$), and the edge set A represents all valid resolution steps ($\mathbb{S}$). A visual representation of this $\mathbb{KG}$ is illustrated in Figure 1.

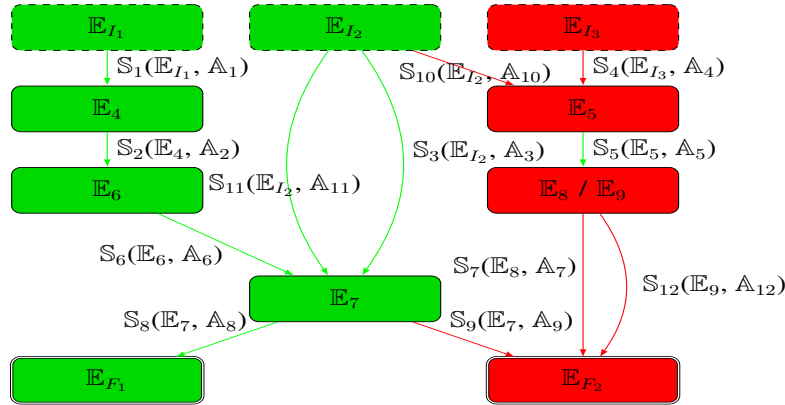


Figure 1. Visual representation of the proposed Knowledge Graph ($\mathbb{KG}$).

Within the $\mathbb{KG}$, each node represents a possible problem state; visually, initial states are distinguished by dashed borders, and final states by double borders. Topologically, a single state may have multiple incoming and outgoing edges. Each directed edge corresponds to a possible step executed by a student via a specific action $\mathbb{A}$. Consequently, a student's resolution $\mathbb{R}$ is formally defined as a path traversing this graph. Furthermore, color-coding is employed to denote pedagogical correctness: green nodes and edges signify correct states and valid steps, respectively, whereas red elements indicate incorrect states and invalid steps. Thus, a correct resolution consists exclusively of green vertices and edges, while an incorrect resolution path contains at least one red component.

To author domain knowledge, a comprehensive, seven-step instantiation process was proposed (Figure 2). In Step 1, the teacher registers the initial problems into the system. Step 2 involves students solving these tasks via step-by-step resolutions; they receive feedback from the ITS, apply it to their problem-solving process, and subsequently evaluate its pedagogical effectiveness. This evaluation data serves as a critical parameter for both the feedback recommendation system and future expert moderation. In Step 3, the system updates the $\mathbb{KG}$ based on the submitted resolutions and feedback evaluations, utilizing them as learning points to map new states, steps, and feedback. This updating phase is executed incrementally as new resolutions are submitted, driven by an algorithm denoted as *updateKG*.

A foundational element of this methodology is the cyclical learning loop established between Steps 2 and 3. The dynamic feedback generated in Step 3 is delivered to students in Step 2, who iteratively assess its efficacy. Consequently, the novel resolutions and ongoing evaluations from Step 2 continuously are used into Step 3 to refine existing knowledge paths. This establishes a robust, incremental authoring architecture

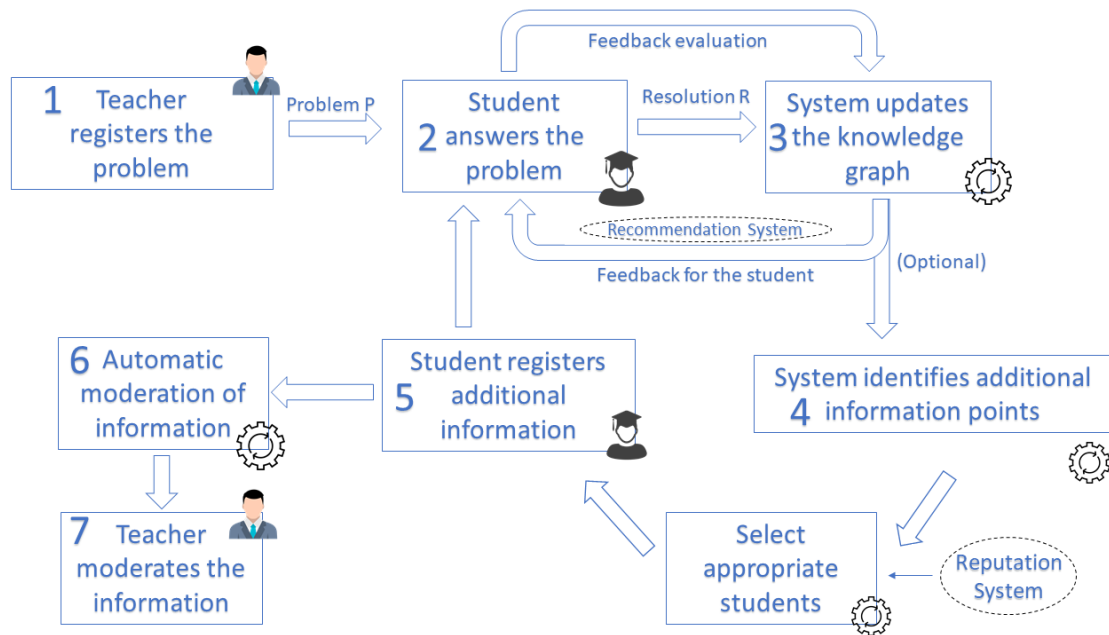


Figure 2. Overview of the Knowledge Graph Authoring Process

with students' Collective Intelligence (CI) at its core. Following this cycle, Step 4 analyzes the current state of the $\mathbb{KG}$ alongside the student's resolution R to identify potential knowledge gaps. The system then automatically ask supplementary explanatory information from top-performing students, guided by a built-in reputation system. Step 5 then consists of registering the additional information from the selected students, thereby improving their reputation. Step 6 introduces an automated moderation layer by the system to identify and eliminate inaccurate, incorrect, or inappropriate information that may be present. Finally, Step 7 consists of manual moderation performed by the teacher with the aim of refining the quality of the information contained in the system to ensure the pedagogical integrity of the system.

To construct the $\mathbb{KG}$ for each ITS problem, an algorithm that iteratively extracts problem-solving patterns from students' solutions was developed. Specifically, the algorithm receives as input a student resolution $\mathbb{R}$ alongside their feedback evaluations, and executes the following procedure:

1. **Graph Retrieval/Initialization:** Retrieve the current knowledge graph associated with problem P . If no graph is registered, initialize a new, empty graph. Alternatively, the instructor may pre-initialize the graph by providing at least one correct reference resolution.
2. **Evaluation Update:** Evaluate and update the parameters of the graph elements (states and steps) corresponding to the received feedback evaluations.
3. **Path Integration:** For each step $\mathbb{S}$ contained in the resolution $\mathbb{R}$:
 - I. Identify the *source state* of $\mathbb{S}$ in the graph. If it does not exist in the graph, instantiate it as a new state node. If $\mathbb{S}$ is the *first step* of $\mathbb{R}$, mark this node as an initial state.
 - II. Identify the *target state* of $\mathbb{S}$ in the graph. If it does not exist in the graph, instantiate it as a new state node. If $\mathbb{S}$ is the *last step* of $\mathbb{R}$, mark this node as a final state.

- III. Locate the corresponding edge in the graph whose source state, target state, and action are identical to step $\mathbb{S}$. If this edge is absent, create it to connect the respective nodes.
4. **Attribute Update:** For each state and step instantiated or retrieved in the preceding phase:
 - I. Update the correctness and validity parameters of all graph elements (nodes and edges) associated with the current resolution $\mathbb{R}$.

The last algorithmic step updates the correctness metrics for the states and steps within the $\mathbb{KG}$. These values are estimated based in the premise that correct and incorrect elements (states or steps) predominantly manifest within correct and incorrect resolutions, respectively. This assumption was subsequently validated empirically with students. Formally, let C_R denote the comprehensive set of resolutions used to construct the $\mathbb{KG}$, with C_{RC} and C_{RI} representing the subsets of correct and incorrect resolutions, respectively. Then the function PE (Equation 1) to indicate whether a specific element is present within a given resolution, and the aggregate function PEC (Equation 2) to quantify the total occurrences of an element across a set of resolutions were defined. Based on these components, the correctness function $C(\mathbb{E}$ or $\mathbb{S})$ was defined, as illustrated in Equation 3. Conceptually, this equation calculates the correctness of an element by subtracting its frequency in correct resolutions from its frequency in incorrect resolutions, normalizing the result dividing by its total appearances across all resolutions. Consequently, the function yields a continuous metric within the interval $[-1, 1]$.

$$PE(E, R) = \begin{cases} 1 & \text{if } E \in R \\ 0 & \text{if } E \notin R \end{cases} \quad (1)$$

$$PEC(E, C_R) = \sum_{i=1}^{|C_R|} PE(E, R_i) \quad (2)$$

$$C(\mathbb{E}) \text{ or } C(\mathbb{S}) = \frac{PEC(E, C_{RC}) - PEC(E, C_{RI})}{PEC(E, C_R)} \quad (3)$$

Finally, the overall correctness of a resolution, denoted as $C(R)$, is defined as the sum of the average correctness of the states E with the average of the validity of the steps S contained in R , divided by 2 to normalize the results, keeping the interval in $C(R) = [-1, 1]$, as presented in Equation 4 and considering C_E the set of states and C_S the set of steps of $\mathbb{R}$.

$$C(\mathbb{R}) = \frac{1}{2 * |C_E|} * \sum_{i=1}^{|C_E|} C(\mathbb{E}_i) + \frac{1}{2 * |C_S|} * \sum_{j=1}^{|C_S|} C(\mathbb{S}_j) \quad (4)$$

Based on these calculations, any given element — whether a state, step, or entire resolution — is provisionally classified as correct if $C \geq 0.8$, and incorrect if $C \leq 0.8$. Subsequently, students are prompted to manually verify and confirm these automated classifications. During this evaluation, the system analyzes each state and step of the student's submitted solution, comparing them against the established KG. Specifically, it identifies elements that exactly match existing graph nodes and edges, as well as those located on alternative paths that are topologically linked to the student's resolution trajectory.

4. Results and Discussions

To evaluate the domain knowledge graphs generated by our approach, we developed an ITS focused on the domain of numerical expressions and applied with 147 students, predominantly comprising undergraduate students from STEM disciplines (mathematics, engineering, and computer science), who were recruited via invitation and voluntarily engaged with the system. Within the ITS environment, each student received brief instructional material regarding numerical expressions before undertaking the primary task. This task consisted of solving one to three problems of increasing difficulty, pre-authored by a domain expert, subject to a strict 10-minute limit per problem.

The students answered each problem using a step-by-step resolution method. During this process, they received system-generated feedback and subsequently evaluated its pedagogical effectiveness, adhering to the continuous authoring cycle detailed in Section 3. Table 1 presents a comprehensive summary of the resolution data. In total, exactly 249 resolutions were recorded across the experiment, with the performance data clearly reflecting the escalating difficulty of the problems:

- **Problem 1 (P_1):** Out of 99 total attempts, 81 (81.8%) were valid resolutions. Among these valid attempts, 56 (69.1%) were correct and 25 (30.9%) were incorrect.
- **Problem 2 (P_2):** Out of 81 total attempts, 72 (88.9%) were valid resolutions. The correctness rate decreased as expected, yielding 41 (56.9%) correct and 31 (43.1%) incorrect resolutions.
- **Problem 3 (P_3):** Out of 69 total attempts, 55 (79.7%) were valid resolutions. Reflecting the highest difficulty level, the correctness rate dropped significantly to 18 (32.7%) correct, with 37 (67.3%) incorrect resolutions.

	Resolution T / I / V	Resolution C / I	Time(s) Min / Max / Avg	Knowledge Avg	# Steps Min / Max / Avg
P_1	99 / 18 / 81	56 / 25	59 / 567 / 236.54	7.86	3 / 12 / 7
P_2	81 / 9 / 72	41 / 31	45 / 590 / 250.50	7.98	4 / 11 / 7
P_3	69 / 14 / 55	18 / 37	79 / 593 / 375.13	8.03	4 / 15 / 8.34

Table 1. Summary of Student Resolution Data

The ITS incrementally updated the knowledge graph via the *updateKG* algorithm after each submitted resolution. To thoroughly analyze this evolutionary process, we focus specifically on the topological growth of problem P_1 . Figure 3(a) illustrates the initial $\mathbb{KG}$, which comprises the correct reference resolution pre-registered by the domain expert. As subsequent student resolutions were processed, the ITS dynamically integrated new states and steps, updated correctness metrics, and branched out novel solution pathways. This early expansion is captured in Figure 3(b), which depicts the P_1 graph after incorporating 12.5% of the total responses.

This structural growth continued continuously with incoming data, as demonstrated in Figure 4, showing the expanded $\mathbb{KG}$ after processing 50% of the student resolutions. Furthermore, these visualizations encode supplementary pedagogical data to facilitate analysis. Node size and edge thickness are scaled proportionally to the students' traversal frequency, allowing for the rapid identification of the most common resolution

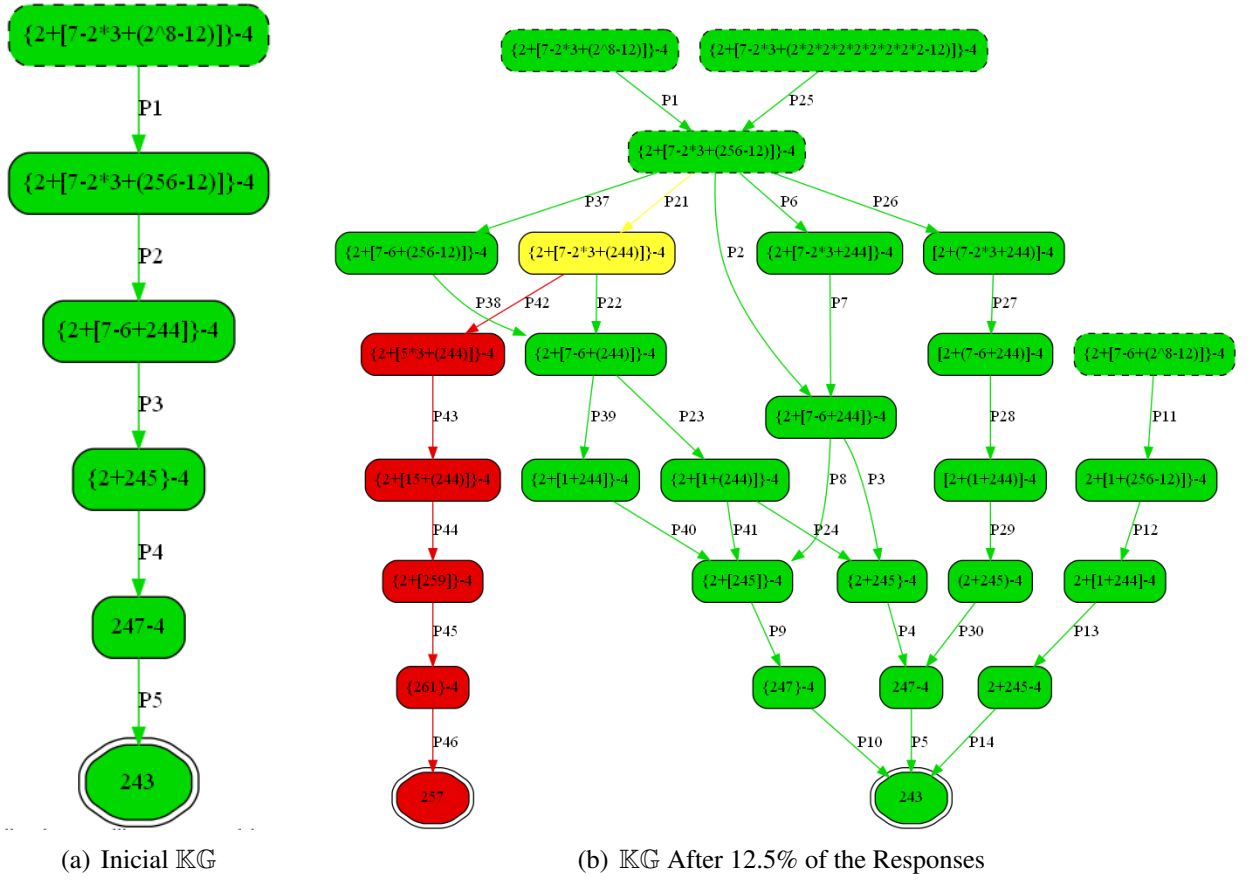


Figure 3. Evolution of the Knowledge Graph ($\mathbb{KG}$) Using the Proposed Authoring Process

pathways (highlighted in blue). Additionally, orange arrows are superimposed to pointing to the most frequent student errors. These are further categorized into either foundational mathematical miscalculations (indicated by points 1 and 3) or deeper conceptual misconceptions (indicated by points 2 and 4).

Table 2 details the structural properties of the generated $\mathbb{KG}$ s, illustrating that the volume of states and steps expanded significantly in direct correlation with task difficulty. This exponential growth underscores the magnitude of the authoring challenge faced by domain experts. By leveraging CI, however, our system dynamically instantiates and updates this information as novel resolutions are submitted, enabling the scalable construction of highly complex $\mathbb{KG}$ s. To rigorously evaluate the efficacy of this approach, three independent domain experts evaluated each generated $\mathbb{KG}$ independently. The resulting classifications, based on the majority of experts, were then compared against our model's automated outputs, yielding the confusion matrices detailed in Table 3.

Based on these matrices, comprehensive statistical metrics were derived, as summarized in Table 4. The data demonstrates that the proposed model achieves robust accuracy in predicting both states and steps across all problems, consistently exceeding 70%. Similarly, precision rates remain exceptionally high, surpassing 80% in all scenarios. This indicates a strong capacity of the model to minimize false positives, thereby ensuring that elements flagged as correct are highly reliable.

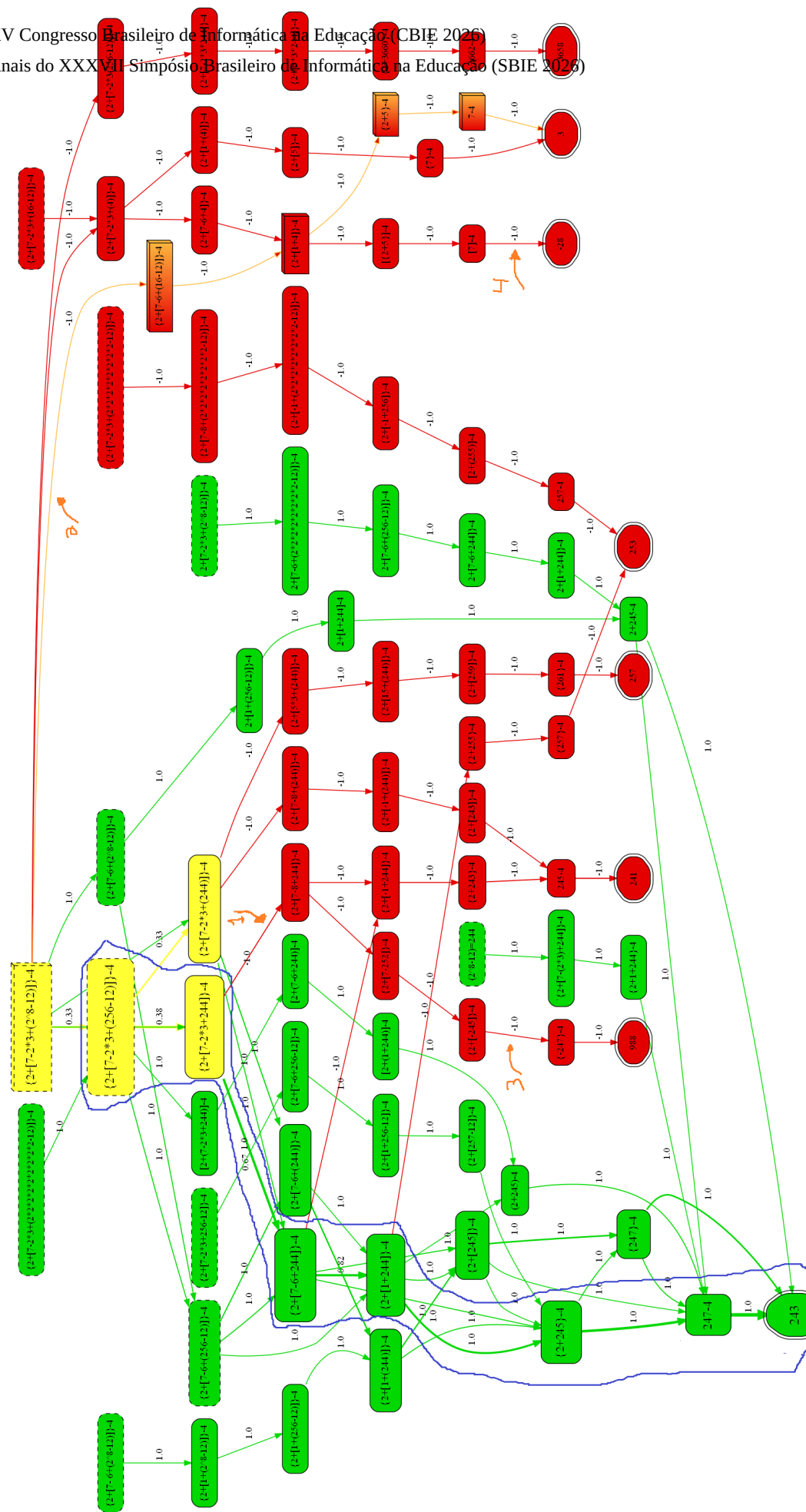


Figure 4. Evolution of the Knowledge Graph (KG) Using the Proposed Authoring Process With 50% of the Answers

	States Tot/Init/Int/Fin	States Cor/Inc/Inconc.	Steps Tot/Val/Inv/Inconc.
P_1	122 / 19 / 90 / 13	42 / 74 / 6	165 / 82 / 82 / 1
P_2	123 / 19 / 90 / 14	27 / 81 / 15	182 / 78 / 99 / 5
P_3	240 / 16 / 192 / 32	37 / 188 / 15	282 / 66 / 211 / 5

Table 2. Summary of the Generated Knowledge Graphs (KGs)

		Expert - States			Expert - Steps	
	State	Correct	Incorrect	Step	Valid	Invalid
P_1	Correct	34	8	Valid	79	3
	Incorrect	1	73	Invalid	0	82
	Inconclusive	6	0	Inconclusive	1	0
P_2	Correct	22	5	Valid	70	8
	Incorrect	8	73	Invalid	16	83
	Inconclusive	15	0	Inconclusive	5	0
P_3	Correct	36	1	Valid	64	2
	Incorrect	50	138	Invalid	61	150
	Inconclusive	15	0	Inconclusive	5	0

Table 3. Confusion Matrix Comparing Model Predictions with Expert Evaluations

Group	ACC Accuracy	PPV Precision	TPR Recall	F1 F-Score	TNR Specificity	FPR type 1	FNR type 2	MCC
P1-States	0,8770	0,8095	0,8292	0,8192	0,9012	0,0988	0,1708	0,7263
P1-Steps	0,9757	0,9634	0,9875	0,9753	0,9647	0,0353	0,0125	0,9518
P2-States	0,7723	0,8148	0,4888	0,6110	0,9358	0,0642	0,5112	0,4943
P2-Steps	0,8406	0,8974	0,7692	0,8284	0,9120	0,088	0,2308	0,6884
P3-States	0,7250	0,9729	0,3564	0,5217	0,9928	0,0072	0,6436	0,4775
P3-Steps	0,7588	0,9696	0,4923	0,6530	0,9868	0,0132	0,5077	0,5641

Table 4. Performance Evaluation Metrics of the Proposed Model

Conversely, we observed lower recall values, most notably in problem P_3 . This decline is corroborated by the false negative rates for steps reported in Table 3 (8.79% in P_2 and 21.63% in P_3), implying that the model occasionally misclassified valid positive states as negative. The F-score metrics demonstrate solid performance for problem P_1 and for the states in problems P_2 and P_3 (all exceeding 0.65). The lowest F-scores were recorded for the steps in problems P_2 and P_3 (0.6110 and 0.5217, respectively); however, performance consistently remained above the 0.50 threshold, establishing the viability of the CI-driven authoring approach even in complex problems.

An analysis of the error metrics reveals consistently low False Positive Rates (FPR), remaining below 10% across all cases for all problems. This demonstrates the model's strong discriminatory capacity to avoid misclassify genuinely negative elements as positive. Consequently, when the system classifies a state as correct or a step as valid, this prediction is accurate over 90% of the time, ensuring a high level of pedagogical reliability. Conversely, an examination of the False Negative Rate (FNR) presents a more

nuanced outcome. While the FNR remains relatively low (under 25%) in the majority of scenarios, notable exceptions exist where values exceed 50%. This elevated FNR indicates that the model exhibits a conservative bias; it occasionally fails to recognize novel, valid student solutions, erroneously classifying them as incorrect.

Finally, the Matthews Correlation Coefficient (MCC) was calculated, a statistical metric that evaluates the quality of binary classifications [Matthews 1975]. The MCC yields a correlation value within the interval $[-1, +1]$, where $+1$ denotes a perfect prediction, 0 indicates a random classification, and -1 signifies total disagreement. It is recognized in the literature as a more reliable and informative metric than standard Accuracy and F1-score [Chicco and Jurman 2020]. The results demonstrate strong MCC performance across all problem groups, with values consistently approaching or exceeding 0.5 , and peaking at 0.95 for problem P_1 .

Based on the statistical analysis presented, **it can be conclude that it is highly feasible to author domain knowledge for ITS inner loops by leveraging student Collective Intelligence (CI), achieving a level of quality and structural integrity equivalent to traditional, expert-driven creation.** When examining the specific states that exhibited divergence between the model and the experts, we found that the majority of discrepancies came from the incorrect application of parentheses. This variance resulted from minor attentional oversights during the manual evaluation of fine-grained elements. Furthermore, we identified a fundamental difference in the evaluation heuristics employed: whereas students tended to assess individual steps holistically, the experts analyzed each step in strict, logical isolation. Reconciling this misalignment in evaluation granularity represents a critical avenue for future refinements of the proposed model.

Furthermore, providing fine-grained, step-by-step resolutions and comprehensive feedback demands significant manual effort from instructors. This creates a severe authoring bottleneck that inherently limits the scalability and practical application of such systems. Historically, the literature estimates that 200 to 300 hours of development are required per single hour of instruction, an overhead that can only be reduced by half using traditional authoring tools like CTAT [Aleven et al. 2006]. To evaluate the efficiency of our approach, we measured the cumulative time required to construct the knowledge graph organically through student interactions. For the evaluated tasks, the students spent a total of 23,425 seconds (approximately 6.5 hours) submitting 99 resolutions for problem P_1 ; 20,192 seconds (5.6 hours) for the 81 resolutions of P_2 ; and 25,379 seconds (7.05 hours) for the 69 resolutions of P_3 . While a strict one-to-one comparison with expert authoring times is challenging due to methodological differences, **the data strongly suggests that generating domain knowledge via CI drastically reduces the overall time compared to traditional expert-driven authoring.** Additionally, continuously crowdsourcing resolutions from students organically bridges the system's inherent knowledge gaps, continuously enriching its capacity to deliver high-quality, adaptive feedback.

Existing literature demonstrates that Collective Intelligence (CI) holds substantial potential within educational technologies to enhance collaboration, interaction, and learning processes, positively impacting both students and instructors [Tenório et al. 2022]. The findings of the present study corroborate this potential, demonstrating that CI can be leveraged to construct the core components of an ITS domain model. This CI-driven approach achieves a level of quality equivalent to expert-authored models but in signif-

icantly less time, thereby substantially mitigating the instructor's authoring bottleneck. Furthermore, student CI can be harnessed for applications far beyond answer validation. The system can automatically identify specific nodes within the knowledge graph that require additional pedagogical support, subsequently prompting students to author explanatory feedback for their peers. This peer-generated feedback can then be continuously evaluated by the students itself. Through this incremental process, the ITS is enabled to automatically filter and recommend only the most highly-rated hints. Ultimately, this paradigm shifts the instructor's role from a primary content creator to a strategic content moderator, liberating valuable time to address complex, individualized student needs.

5. Conclusions and Future Works

This study proposed and validated a novel, CI-driven framework designed to mitigate the persistent authoring bottleneck in Intelligent Tutoring Systems (ITS). By leveraging the Collective Intelligence of students to incrementally construct and update domain Knowledge Graphs ($\mathbb{KG}$), we demonstrated that it is highly feasible to author fine-grained ITS inner loops without relying exclusively on domain experts. Our empirical evaluation, conducted with 147 STEM students, confirms that the CI-generated domain models achieve a level of structural integrity equivalent to traditional, expert-driven creation, but at a fraction of the time. The statistical analysis shows that the system demonstrated strong predictive capabilities, consistently maintaining accuracy rates above 70% and precision above 80%, alongside good Matthews Correlation Coefficient (MCC) scores. The analysis also revealed a conservative bias in the model's classification — evidenced by a higher False Negative Rate (FNR) in more complex problems — ensuring a low rate of false positives and, consequently, a high degree of pedagogical reliability. Despite these promising results, the evaluation highlighted critical avenues for refinement. We observed divergences between the model and expert, primarily stemming from the incorrect application of parentheses and minor attentional oversights. Addressing the misalignment in evaluation granularity constitutes a primary focus for our immediate future work. Specifically, we intend to refine the model's algorithms to better reconcile the contextual versus isolated evaluation heuristics observed between students and experts.

Furthermore, we plan to expand the application of student CI beyond structural authoring to encompass the generation of rich pedagogical interventions. By automatically identifying fragile nodes within the $\mathbb{KG}$, the ITS will dynamically ask to students to author support materials, such as contextual hints, conceptual explanations, justifications, and specific error feedback. Finally, to establish the validity and generalization of the proposed framework, future studies will deploy the CI-driven ITS within authentic classroom environments and across diverse educational domains. Conducting these in-the-wild experiments will be crucial for empirically quantifying the tangible learning gains for students, as well as the practical reduction of authoring overhead for educators in everyday practice. Finally, while the descriptive metrics and correlation coefficients strongly indicate the model's efficacy, the current analysis does not include inferential statistics or confidence intervals. Consequently, although the observed trends are highly promising and validate the feasibility of the CI-driven authoring process, the absence of formal significance testing limits the statistical generalization of these specific quantitative outcomes to broader populations.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Declaration of AI usage

During the preparation of this manuscript, the authors used Gemini exclusively to refine the english grammar and textual fluency. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

- Aleven, V., McLaren, B. M., Sewall, J., and Koedinger, K. R. (2006). The cognitive tutor authoring tools (ctat): preliminary evaluation of efficiency gains. In *International Conference on Intelligent Tutoring Systems*, pages 61–70. Springer.
- Aleven, V., McLaren, B. M., Sewall, J., and Koedinger, K. R. (2009). A new paradigm for intelligent tutoring systems: Example-tracing tutors. *International Journal of Artificial Intelligence in Education*, 19(2):105–154.
- Chicco, D. and Jurman, G. (2020). The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1).
- Cross, A., Bayyapunedi, M., Ravindran, D., Cutrell, E., and Thies, W. (2014). Vidwiki: Enabling the crowd to improve the legibility of online educational videos. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*, page 1167–1175, New York, NY, USA.
- Heffernan, N. T. and Heffernan, C. L. (2014). The assistments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education*, 24(4):470–497.
- Heffernan, N. T., Ostrow, K. S., Kelly, K., Selent, D., Van Inwegen, E. G., Xiong, X., and Williams, J. J. (2016). The future of adaptive learning: Does the crowd hold the key? *International Journal of Artificial Intelligence in Education*, 26(2):615–644.
- Leimeister, J. M. (2010). Collective intelligence. *Business & Information Systems Engineering*, 2(4):245–248.
- MacLellan, C. J., Koedinger, K. R., and Matsuda, N. (2014). Authoring tutors with sim-student: An evaluation of efficiency and model quality. In *International conference on intelligent tutoring systems*, pages 551–560. Springer.
- Matthews, B. (1975). Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442–451.
- Meza, J., Jimenez, A., Mendoza, K., and Vaca-Cárdenas, L. (2018). Collective intelligence education, enhancing the collaborative learning. In *International Conference on eDemocracy & eGovernment (ICEDEG)*, pages 24–30. IEEE.

- Mitrovic, A. (2003). An intelligent sql tutor on the web. *International Journal of Artificial Intelligence in Education*, 13(2-4):173–197.
- Nkambou, R., Mizoguchi, R., and Bourdeau, J. (2010). *Advances in intelligent tutoring systems*, volume 308. Springer Science & Business Media.
- Pane, J. F., Griffin, B. A., McCaffrey, D. F., and Karam, R. (2014). Effectiveness of cognitive tutor algebra i. *Educational Evaluation and Policy Analysis*, 36(2):127–144.
- Polson, M. C. and Richardson, J. J. (2013). *Foundations of intelligent tutoring systems*. Psychology Press.
- Psotka, J., Massey, L. D., and Mutter, S. A. (1988). *Intelligent tutoring systems: Lessons learned*. Psychology Press.
- Razzaq, R., Ostrow, K. S., and Heffernan, N. T. (2020). Effect of immediate feedback on math achievement at the high school level. In *Artificial Intelligence in Education*, pages 263–267.
- Schulze, K. G., Shelby, R. N., Treacy, D. J., Wintersgill, M. C., Vanlehn, K., and Gertner, A. (2000). Andes: An intelligent tutor for classical physics. *Journal of Electronic Publishing*, 6(1).
- Seffrin, H., Rubi, G., and Jaques, P. (2012). O modelo cognitivo do sistema tutor inteligente pat2math. In *Simpósio Brasileiro de Informática na Educação-SBIE*.
- Sun, C., Li, S.-w., and Lin, L. (2016). Thread structure prediction for mooc discussion forum. In *Social Computing*, pages 92–101, Singapore. Springer Singapore.
- Tacoma, S., Drijvers, P., and Jeuring, J. (2021). Combined inner and outer loop feedback in an intelligent tutoring system for statistics in higher education. *Journal of Computer Assisted Learning*, 37(2):319–332.
- Tenório, T., Isotani, S., and Bittencourt, I. I. (2022). Authoring inner loops of intelligent tutoring systems using collective intelligence. In *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners' and Doctoral Consortium: 23rd International Conference, AIED 2022, Durham, UK, July 27–31, 2022, Proceedings, Part II*, page 400–404, Berlin, Heidelberg. Springer-Verlag.
- Tenório, T., Isotani, S., Bittencourt, I. I., and Lu, Y. (2021). The state-of-the-art on collective intelligence in online educational technologies. *IEEE Transactions on Learning Technologies*, 14(2):257–271.
- Tenório, T., Isotani, S., and Bittencourt, I. I. (2020). Inteligência coletiva como ferramenta de apoio na construção de loops internos em sistemas tutores inteligentes. In *Anais do XXXI Simpósio Brasileiro de Informática na Educação*, pages 1233–1242, Porto Alegre, RS, Brasil. SBC.
- Tenório, T., Rodrigues, L., Isotani, S., and Bittencourt, I. (2023). Empowering instructors with collective intelligence: Learning problem-solving paths to facilitate feedback generation. In *Anais do XXXIV Simpósio Brasileiro de Informática na Educação*, pages 1249–1259, Porto Alegre, RS, Brasil. SBC.
- Vaish, R., Goyal, S., Saberi, A., and Goel, S. (2018). Creating crowdsourced research talks at scale. In *Proceedings of the 2018 World Wide Web Conference*, page 1–11,

Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

VanLehn, K. (2006). The behavior of tutoring systems. *International journal of artificial intelligence in education*, 16(3):227–265.

VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221.

Weitekamp, D., Harpstead, E., and Koedinger, K. R. (2020). An interaction design for machine teaching to develop ai tutors. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pages 1–11.

Zaidan, O. F. and Callison-Burch, C. (2011). Crowdsourcing translation: Professional quality from non-professionals. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 1220–1229. Association for Computational Linguistics.