Design, Development and Evaluation of a Lightweight Knowledge-based System for Theoretically-grounded Math Error Classification

Anderson P. Avila-Santos[1,2], Luiz Rodrigues[1], Thomaz E. V. da Silva[1], Thales Vieira[1], Marcelo Marinho[3], Valmir Macario[3], Diego Dermeval[1,5], Ig Ibert Bittencourt[1,5], Seiji Isotani[4,5]

[1]Center for Excellence in Social Technology - Federal University of Alagoas
Maceió - AL - Brazil

[2]Computing Department - State University of Londrina - Londrina - PR - Brazil

[3]Federal Rural University of Pernambuco - Recife - PE - Brazil

[4]Institute of Mathematical and Computer Sciences - University of São Paulo
São Carlos - SP - Brazil

[5]Graduate School of Education - Harvard University - USA

luiz.rodrigues@nees.ufal.br

Abstract. *Mathematical problem solving is essential for developing analytical skills and critical thinking among learners. Traditional methods of detecting and correcting errors in mathematical problem-solving are manual and time-consuming, leading to delayed feedback, which can hinder effective learning. Intelligent Tutoring Systems (ITS) provide real-time, error-specific feedback but require direct interaction with computational devices, limiting their use in classrooms with insufficient technological infrastructure. ITS unplugged (ITS-U) have been proposed to address these technological barriers, but no research has investigated how to design a lightweight error classificatin systems aligned to the resource-constrained setting of ITS-U. Therefore, this paper presents the Mathematical Solutions Error Classification (MSEC) API, a lightweight, knowledge-based tool designed to automate the classification of errors in mathematical solutions. This API categorizes errors into theoretically-grounded types, such as syntactical mistakes, conceptual misunderstandings, and calculation errors. MSEC is particularly suited for low-tech educational environments, aligning with the principles of ITS-U. We implemented MSEC within an ITS-U and conducted a case study involving 49 students, demonstrating its practical applicability and positive contribution to teaching and learning. The study highlights MSEC's efficiency, ability to provide real-time feedback, and adaptability to various educational contexts, offering a significant advancement in automated error detection for ITS-U.*

1. Introduction

Mathematical problem solving not only contributes to improve analytical skills but to develop critical thinking among learners [Polya 2014, Levin and Levin 2012]. Traditionally, detecting and correcting errors in mathematical problem-solving are manual, time-consuming tasks that educators invest significant time in identifying mistakes in student

DOI: 10.5753/sbie.2024.242500

work, often resulting in delayed feedback, while timely feedback is necessary for effective learning [Shute 2008]. Hence, this manual process limits the frequency and the readiness of feedback, which are vital for learning progress [Hattie and Timperley 2007].

To address these challenges, automated solutions have emerged as a significant advancement. Intelligent Tutoring Systems (ITS), for instance, are widely known for their potential to provide error-specific feedback in real time [Nkambou et al. 2010, VanLehn 2006], which is often associated with ITS's positive impact on learning outcomes compared to other instructional methods and digital tools [Hillmayr et al. 2020, Steenbergen-Hu and Cooper 2014]. Despite these benefits, ITSs are educational technologies that demand a direct interaction between the learner and a computational device [Mousavinasab et al. 2021], restricting its widespread adoption in classrooms with limited technological infrastructure [Isotani et al. 2023].

On the other hand, many global south countries, such as Brazil, face issues related to technological infrastructure [Gasevic et al. 2018]. These limitations contribute to widening learning inequalities, as learners from underserved regions are restricted from benefiting from the solutions adopted when technological infrastructure is available, contrasting to Sustainable Development Goal 10: *Reduce inequality within and among countries* [Lin et al. 2023, Vinuesa et al. 2020]. These issues led to the proposal of Artificial Intelligence in Education (AIED) Unplugged, a paradigm aimed to address these issues by calling for research and development on AIED systems tailored to resource-restricted settings [Isotani et al. 2023].

Given ITSs prominence on AIED research, AIED Unplugged (AIED-U) motivated the conceptualization of a framework for ITS unplugged (ITS-U) [Veloso et al. 2023]. Considering the unavailability of digital devices (e.g., computers, smartphones, etc.) for each student, the framework explores one smartphone's availability to propose the following flow. First, students solve math questions on paper sheets, then, the teacher uses a mobile application to photograph students' solutions. In doing so, the application should be able to transcribe the equations within the picture and assess each solution to provide error-specific feedback in real-time [Rodrigues et al. 2023]. Hence, teachers act as a proxy of the ITS-U: they see the feedback on the mobile application and provide them to students, as advocated by AIED-U's proxy principle [Isotani et al. 2023]. Consequently, while the proposed solution involves multiple technological challenges, such as equation detection and handwritten math equation recognition, error classification is a key step to enable the system to provide error-specific feedback.

Previous research has discussed AIED systems that relate to ITS-U's ideal, but are limited in terms of error classification. [Davis et al. 2020] introduces Homework Helper, a mobile application aimed to help students perform arithmetic homework, presenting an algorithm to identify addition errors in 2D equations. However, their error identification is limited to addition, besides presenting no theoretical ground for the error types. [Patel et al. 2022] introduces a framework and a prototype for a similar app, but concerning algebra assessments. Similarly, the study presents no theoretical grounding for the error classification system, using an ad-hoc analysis to tabulate error patterns based on a small sample of students solutions. Finally, [Rodrigues et al. 2024] presents and evaluates a high-fidelity prototype of MathAIde, an ITS-U that aims to aid numeracy education. Nevertheless, as the study is focused on the usability of the solution, it concerns the interaction flow, not discussing how its error classification works. Furthermore, while recent advancements on Large Language Models (LLMs) raise the potential to explore them for such tasks, their resource-intensive nature (e.g., demanding a GPT to answer user prompts) limits its adoption in unplugged contexts.

Reviewing related work demonstrates that there is a practical knowledge gap in terms of a lightweight, theoretically grounded system able to perform error-specific classification in math equation for ITS-U. Therefore, this paper introduces the Mathematical Solutions Error Classification (MSEC) API, a tool designed to streamline the process of error detection in educational environments. The MSEC API categorizes errors into theoretically-grounded types, such as syntactical mistakes, conceptual misunderstandings, and calculation errors, among others [Conati and Merten 2014]. By automating the classification of these error types, the API enables educational technology like ITS-U to provide more targeted and immediate feedback, empowering educators in enhancing their teaching practices with personalized feedback.

Notably, MSEC relies on a knowledge-based approach [Nkambou et al. 2010] to assess and classify math errors types, yielding a lightweight approach suitable to ITS-U compared to, for example, LLMs. Moreover, MSEC identifies errors defined by math experts, which designed errors in light of the Brazilian National Curriculum (Base Nacional Comum Curricular - BNCC) [Santos et al. 2018], enabling MSEC to aid educators and students according to their particular educational needs. Thus, the MSEC API represents a technological leap, offering a digital means to instantly classify errors in mathematical solutions provided by students from a theoretically relevant perspective, while requiring little computational resources, as needed by AIED-U systems.

This study aligns with the principles of Design Science Research (DSR) [Hevner et al. 2004], which emphasizes creating and evaluating innovative artifacts to solve identified problems. Initially, we conducted a requirement-gathering phase to identify specific needs and challenges in mathematical problem-solving. Following this, we designed and developed the MSEC API, employing a microservices architecture to enhance scalability and flexibility. We then integrated the API into an ITS-U and deployed it in real classroom settings, leading to a case study using the MSEC API in a real educational context. Accordingly, this paper reviews the API performance, identifies failure cases, and discusses subsequent adjustments made to improve the API based on the findings from the case study. This iterative evaluation and refinement process highlights the API's practical applicability and provides valuable insights into its impact on teaching and learning processes. Therefore, the main contributions of this paper are i) the development of the MSEC API for error classification in mathematical solutions and ii) a case study that demonstrates the implementation and use of the MSEC API in real educational environments.

2. Method

This research is based on the guidelines of the DSR [Hevner et al. 2004]. The main objective was to design, develop, and evaluate MSEC, a lightweight error classification system (our artifact) aimed to be integrated with educational technology so that it empowers math education and learning with error-specific feedback (interaction with the environment. Consequently, MSEC contributes to the challenge of identifying and providing error-specific feedback for math exercises, particularly for resource-constrained contexts, such as AIED unplugged systems, due to its lightweight approach.

This study involved the following steps. First, we engaged in a cycle of requirements elicitation and engineering with math experts to ensure MSEC is backgrounded by theoretical aspects of relevance for math education so we could design a solution aligned to this domain's needs (see Section 2.1). Second, we developed MSEC following the math experts' insights, while still ensuring it was based on a lightweight approach and compatible with a varied of educational technologies, especially those targeting resource-constrained contexts, such as ITS-U (see Section 2.2). Third, we instantiated the MSEC API within an ITS-U and conducted a case study. This allowed us to demonstrate MSEC's suitability to be integrated with existing educational technology, evaluate its performance and behavior in a real learning context, and, thus, iteratively refine its functioning.

By following the DSR guidelines, we systematically addressed the problem domain through iterative cycles of design, development, and evaluation. DSR enhanced this study by providing a structured approach to incorporate theoretical insights from math education into the practical development of MSEC, ensuring that the solution is both theoretically sound and practically viable. Additionally, the iterative nature of DSR allowed for continuous refinement and validation of MSEC in real-world educational settings, leading to a robust and adaptable solution that meets the needs of diverse educational contexts, as detailed next.

2.1. Brazilian National Curriculum (BNCC)

Following math experts recommendations, MSEC considers the skills presented in the Table 1. Those skills originate from a structured educational framework designed to guide learning progressions in mathematics, the Brazilian National Curriculum (BNCC), which outlines the learning objectives and competencies that students are expected to develop at various stages of their educational journey [Brasil. Ministério da Educação 2018].

For instance, EF01MA06 focuses on constructing and utilizing fundamental addition facts, a skill essential for the foundational understanding of arithmetic operations, which are crucial for problem-solving in everyday contexts. Similarly, EF01MA08 encompasses the development and application of addition and subtraction in problem-solving situations, encouraging students to employ personal strategies and manipulative materials to comprehend mathematical concepts tangibly. As students progress, skills like EF02MA05 and EF02MA06 advance the complexity of these operations, promoting the use of mental and written calculation methods to solve problems involving larger numbers and introducing them to multi-step computations. Skills such as EF01MA02 are aimed at fostering exact or approximate counting abilities, leveraging strategies like pairing and grouping to foster a solid understanding of number sense. Similarly, EF02MA03 is about composing and decomposing numbers, which is instrumental in building flexibility in number usage and an understanding of the place value system.

Regarding multiplication and division, EF03MA03 and EF03MA07 aim at establishing basic facts of multiplication and utilizing these for calculations, as well as solving division problems that incorporate real-world scenarios of sharing and measurement. These skills serve not only as a foundation for advanced mathematical concepts but background a learner's ability to apply mathematical reasoning in everyday life. Therefore,

Table 1. BNCC skills encompassed by MSEC API.

Skill Code	Description
EF01MA06	Construct fundamental addition facts and use them in calculation procedures to solve problems.
EF01MA08	Solve and formulate problem situations involving addition and subtraction with meanings of joining, adding, separating, and taking away, with the support of images and/or manipulative materials, using personal strategies and recording methods.
EF02MA05	Construct basic addition and subtraction facts and use them in mental or written calculation.
EF02MA06	Solve and formulate problem situations involving addition and subtraction with numbers up to three orders of magnitude, with meanings of joining, adding, separating, taking away, using personal or conventional strategies.
EF01MA02	Count in an exact or approximate manner, using different strategies such as pairing and other groupings.
EF02MA03	Compose and decompose natural numbers up to three orders of magnitude, with the support of manipulative materials, through different additions.
EF03MA03	Construct and use basic addition and multiplication facts for mental or written calculation.
EF03MA07	Solve and formulate division problems where the divisor has at most two digits, involving meanings of equal sharing and measurement, using various strategies such as estimation, mental calculation, and algorithms.

MSEC encompasses these skills considering they form an integral part of a comprehensive educational strategy aimed at equipping students with the mathematical knowledge and problem-solving capabilities that are essential for academic success and practical life beyond the classroom.

Based on the selected skills, our team of math experts performed a throughout analysis of students to establish a comprehensive list of possible errors. Initially, all student responses were manually graded by a panel of experts in mathematics education and curriculum, ensuring an in-depth qualitative assessment of students' reasoning and specific errors made. Manual grading also allows experts to consider the reasoning behind students' answers, providing a richer understanding of the cognitive processes at play. During this process, the experts identified the type of error made and established an error taxonomy, developed based on the complexity and nature of the errors, allowing for a systematic classification that encompassed conceptual, procedural, and reading/interpretation-related errors. Table 1 presents these errors, along with their corresponding descriptions, which are aimed to provide educators with a valuable resource to enhance instructional methods and support student learning effectively once identified by MSEC.

Table 2. Error Codes and Descriptions

Error Code	Error Description
D001	Incorrect Sum
C002	Incorrect Counting
V001	Incorrect Visual Representation
NR001	Question Not Resolved
C003	Incorrect Representation
EE001	Mirrored Writing Error
EC001	Comparison Error
EI001	Interpretation Error
M001	Incorrect Multiplication
S001	Incorrect Subtraction
D003	Operation Inversion
Q001	Incorrect Division (Quotient)

2.2. MSEC Error Classification

According to the expert-defined error taxonomy (see Table 2), MSEC focuses on classifying errors in the mathematical operations of addition, subtraction, division, and multiplication. Given a student solution (in LaTeX format) for an exercise concerning the selected skills (see 1) and the exercise's expected answer, MSEC analyses those to determine the solution's correctness or to generate a list containing the mistakes the student made. This setup is based on math experts' recommendations, which highlight students might commit multiple errors while solving a single exercise.

In that context, MSEC error classification is based on three main steps. First, *element extraction*, where MSEC analyzes the equation to extract numbers, operators, results, and carry-overs using regular expressions. For instance, if the student solution is `\\overset { 1 } { 3 } \\overset { 1 } { 0 } \\overset { 1 } { 0 } + 7 3 = 3 2 7`, this step's output is numbers = [30, 73], operator = '+', result = 327, and carry_overs = [1, 1, 1]. Once all elements have been extracted, MSEC's second step is *error identification*. This step uses error one identification method for each possible error (also known as specialist systems), which are detailed next. Finally, the last step is *result return*, wherein MSEC returns a list of all identified errors.

For *Incorrect Sum* (D001), MSEC verifies the accuracy of digit-by-digit calculations, including the propagation of carry-overs. The specialist system checks each digit of the operands and ensures the addition is carried out correctly with proper handling of carry-overs. For instance, if the student solution to "125 + 98" is `\overset11\overset125 + 9 5 = 4 2 0`, MSEC will identify that although the carry-overs are correctly noted, the final sum is incorrect. The correct addition should result in 223, not 420, indicating that the digits are incorrectly added and the carry-over is not properly applied, resulting in an incorrect sum.

In the case of *Incorrect Subtraction* (S001), MSEC checks whether the subtraction operation between the numbers is correctly performed. This involves verifying each digit and ensuring proper borrowing where necessary. For instance, if the student solution to "125 - 98" is `1 2 5 - 9 8 = 3 7`, MSEC will detect that the borrowing is incorrect

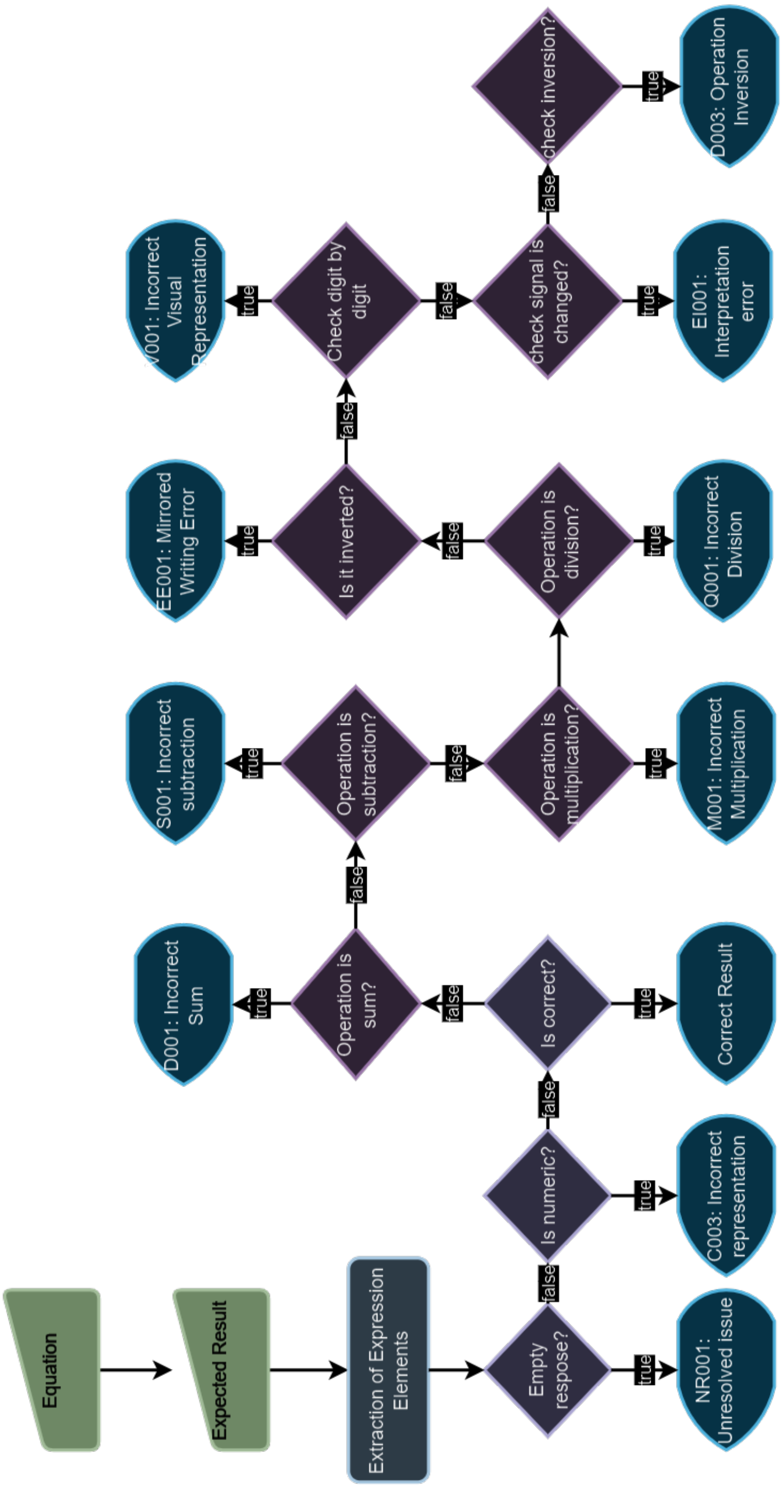

Figure 1. MSEC Error Classification Pipeline. The figure illustrates the process of classifying errors in mathematical solutions using MSEC, which includes three main steps: element extraction, error identification, and result return. Examples of identified errors include incorrect sums, incorrect subtractions, incorrect multiplications, incorrect divisions, incorrect numerical representations, unresolved questions, mirrored writing errors, interpretation errors, and operation inversions.

and the resultant digits do not align with the correct subtraction, marking it as an error.

For *Incorrect Multiplication* (M001), MSEC ensures that the multiplication process is accurately performed for each digit. The system verifies the result of the multiplication. For instance, if the student solution to "12 * 3" is `1 2 * 3 = 4 0`, MSEC will identify that the multiplication steps are incorrect, leading to an erroneous result.

When addressing *Incorrect Division* (Q001), MSEC evaluates the division operation, checking both the quotient and the remainder. It ensures that the division is correctly executed and the remainder is accurately calculated. For instance, if the student solution to "15 / 2" is `1 5 / 2 = 8`, MSEC will detect that the quotient is incorrect, flagging it as a division error. It also checks that the denominator is not zero.

For *Incorrect Representation* (C003), MSEC checks the numerical representation used in the solution. It verifies whether the numbers are correctly formatted and appropriately placed. For instance, if the student solution to "123" is represented with non-numeric values like `one two three` instead of numerical digits, MSEC will detect that the numerical representation is incorrect, marking it as a representation error.

In the case of *Question Not Resolved* (NR001), MSEC checks if the student has provided any solution. If no attempt is made or the solution is left blank, it flags the response as "Question Not Resolved". For instance, if the student does not write any solution for a given problem, MSEC will mark it as NR001.

For *Mirrored Writing Error* (EE001), MSEC identifies instances where the student writes numbers or operations in a mirrored or reversed fashion. For instance, if the student solution to "21 - 12" is `12 - 21 = -9`, MSEC will detect that the numbers are written in a mirrored or reversed way, marking it as a mirrored writing error. The correct solution should maintain the original order of the numbers: $21 - 12 = 9$.

Regarding *Interpretation Error* (EI001), MSEC checks if the student correctly interprets the mathematical operations or problem statements. It ensures the logic behind the solution matches the expected mathematical interpretation. For instance, if the student solution to "4 * 10" is `4 + 1 0 = 1 4`, MSEC will detect that the student has misinterpreted the multiplication as addition, marking it as an interpretation error.

Lastly, for *Operation Inversion* (D003), MSEC identifies if the student has mistakenly inverted the operation. For instance, if the student solution to "12 - 5" is `5 - 12 = -7`, MSEC will detect the inversion of the operation, flagging it as an error.

2.3. MSEC API

We chose a microservices architecture for MSEC API to enhance its scalability, flexibility, and maintenance efficiency. This structure supports the creation of services that operate independently yet work together seamlessly, allowing each component to be updated or scaled without affecting the whole system. Accordingly, this architecture facilitates expanding MSEC features by, for instance, adding new specialist systems, as well as facilitates its integration with various web-based educational technologies.

In MSEC's API, the endpoint `/classify-error` represents the gateway for users to submit their input and receive a detailed error analysis. It accepts POST requests containing data with the student's solution (a math expression) and the expected answer of the exercise the students just solved. As MSEC relies on regular expressions for equation parsing, both parameters are expected as strings. Then, after performing the error classification procedure (see Section 2.2), the API returns the list of detected errors, identifying each of them according to the codes shown in Table 2.

Thank to this approach, MSEC might be easily integrated with several online learning environments. Any web-based system, such as ITSs, virtual learning environments, and gamified systems, might act as a client that consumes MSEC's API to assess the correctness of math solutions. As long as those systems provide the API with its POST parameters, they will be able to receive detailed, theoretically-grounded feedback of the solution's correctness based on MSEC's lightweight, knowledge-based approach.

3. Evaluation

This section presents a few usage cases of MSEC API to exemplify how it receives and processes equations to identify possible errors. Next, the section presents a case study where we integrated the MSEC API into a ITS unplugged, which was deployed into real classrooms, to evaluate its capabilities of identifying math errors.

3.1. Usage Samples

Consider the skill EF02MA06, which involves solving and formulating problem situations requiring addition and subtraction with numbers up to three digits, using personal or conventional strategies. The students were tasked with solving the following problem: ”A school bus transports 125 students in the morning and 98 students in the afternoon. How many students are transported in total during the day?” Then, a given student provided the following response:

$$\begin{array}{r} \overset{1}{1}\overset{1}{2}5 \\ +\ 95 \\ \hline 420 \end{array}$$

In the LaTex format, as expected by MSEC, the solution reads as `\overset{1}{1}\overset{1}{2}5 + 9 5 = 4 2 0`, whereas the expected answer was 223. In this specific instance, the API returned the following errors: D001: Incorrect Addition, V001: Incorrect Visual Representation. The addition is incorrect as the final value does not match the expected result. Additionally, the visual representation provided by the student is incorrect, as they wrote the number 5 instead of 8 in 98.

As another example, consider skill EF03MA07, which involves solving and formulating division problems where the divisor has at most two digits, involving meanings of equal sharing and measurement, using various strategies such as estimation, mental calculation, and algorithms. The following problem was developed targeting this skill: ”A farmer has 4 bags of corn. Each bag contains 10 ears of corn. How many ears of corn does the farmer have in total?” The expected answer for this question was 40. In this case, the student responded as follows: `4 + 1 0 = 1 4`, and the API returned EI001: Interpretation Error, given that the student performed an addition operation where a multiplication operation was required.

These examples illustrate the capability of the MSEC API to accurately identify and classify errors in student responses to mathematical problems. By providing detailed feedback on specific error types, the API supports understanding and learning, helping students to recognize and correct their mistakes effectively.

3.2. Case Study Setting

To further evaluate MSEC, we instantiate it in an ITS-U and conducted a case study. This case study involved two teachers, which voluntarily made themselves available to use MathAIde in their classrooms following the researchers' invitation. This procedure was reviewed and approved by the Research Ethics Committee (CAAE: 73285823.2.0000.5390).

Both teachers had previous experience with MathAIde, and each of them used the ITS in four lessons. The teachers were a man and a woman, aged between 25 and 29 and between 50 and 54, respectively, with 6 to 10 years of teaching experience. While the man was from a small, sourthern Brazilian city, the woman was from a large, northern Brazilian city. Both Teachers used the application with their 4th and 5th years classes, totaling 49 students. In addition to the Informed Consent Form applied to teachers, we collected consent from the schools where the two teachers work and the parents of each student who participated in this study, ensuring that all ethical aspects were considered.

The case study procedure was as follows. First, MathAIde was made available for teachers. Second, each teacher utilized MathAIde during their classes over four different days. During each session, teachers applied exercise lists to their students and recorded each student's answers. In preparation for each class, teachers printed the exercise lists, where the first two concerned skill EF02MA06 and the remainder concerned skill EF03MA07. Lastly, teachers executed the lessons, applying the mentioned lists, where all captures and records they made during the study were stored in the application database to enable subsequent analysis. Note that capturing solutions with MathAIde involves taking a picture of the paper sheet where the student solve the exercise, which is then transcribed and assessed by the ITS-U. This assessment step is where MSEC API was used, following the standard procedure of ITS-U, as discussed in Section 1.

In total, 1179 student answers were collected during the case study. All of those were classified by the MSEC API and, subsequently, forwarded to a researcher experienced in math education for review. This review provided insights into the correctness of the MSEC classifications, revealing interesting real-world scenarios that it failed to properly classify students' equations, as discussed next.

3.3. Lessons Learned

There are three main lessons learned from evaluating MSEC API. First, the case study revealed MSEC's efficiency. Due to its lightweight nature, employing a knowledge-based approach, MSEC was able to perform real-time classifications. Additionally, the expert review of MSEC classification confirmed its ability to successfully classify student errors for the selected BNCC skills, which corroborates its knowledge-based approach informed by the detailed analysis of students solutions our team of math experts conducted. Thereby, these insights demonstrate MSEC's value based on empirical evidence generated following its instantiation in a ITS-U.

Second, instantiating MSEC helped us understand some important limitations. Despite MSEC correctly classified most solutions, we identified important edge cases that MSEC was not prepared to handle (6% of the 1179 solutions). For instance, the API received solutions such as `6 0` and `\\overset { 1 } { 1 } \\overset { 1 } { 5 } 6 + 0 8 9 = 2 4 5`, which were beyond the initial expectation of the specialist systems for error identification. Thanks to this, we were able to improve MSEC's logic, ensuring its ability to cover a broader spectrum of possibilities. For example, we adjusted the system to better handle cases where the carry-over in addition problems was not properly accounted for, improving its accuracy in these scenarios. Therefore, these insights were prominent to further understand and expand MSEC capabilities.

Third, the iterative feedback from educators was of utmost importance in designing and refining MSEC. Involving teachers since MSEC's initial design was of utmost importance to ensure that it is theoretically grounded in BNCC and that it identifies error types of relevance for student learning. Additionally, including teachers in the iterative refinement of MSEC was prominent to enable it to encompass cases that were not predicted at first, as well as to adjust its error codes. Initially, teachers reported that while the API error were insightful, the error messages were sometimes too technical. In response, the error codes were rephrased to be more instructional, providing clear guidance for teachers to help students could correct their mistakes. This user-centered approach not only improved the API's functionality but also made it a more effective tool for enhancing mathematical learning in the classroom.

4. Discussion

The MSEC API represents a significant advancement in the domain of educational technology, specifically targeting the automation of error detection and classification in mathematical problem-solving. Traditional methods of detecting and correcting errors have been manual, requiring significant time and effort from educators, which often resulted in delayed feedback to students. The introduction of the MSEC API automates this process, providing immediate, error-specific feedback that is crucial for effective learning and timely pedagogical intervention when needed.

By categorizing errors into syntactical mistakes, conceptual misunderstandings, and calculation errors, the MSEC API not only identifies errors but also helps educators understand the underlying issues in students' mathematical reasoning. This immediate and targeted feedback empowers teachers to provide personalized instruction, which is critical for addressing individual learning needs and promoting deeper understanding of mathematical concepts. Furthermore, the API's integration with a mobile application allows it to be used in low-tech classrooms, broadening its applicability in diverse educational settings, particularly in resource-restricted environments.

In this sense, the feedback from educators and students has been instrumental in refining the MSEC API. During the case study, teachers noted that while the API provided valuable insights into students' errors, the initial error messages were too technical and not easily understood by students. In response to this feedback, the error messages were rephrased to be more instructional, offering clear guidance on how students could correct their mistakes. This adjustment improved the usability of the API, making it a more effective and attractive tool for enhancing mathematical learning in the classroom.

Despite its significant contributions, the current version of the MSEC API has certain limitations. One major challenge identified during the case study was the API's occasional misclassification of errors, particularly in more complex mathematical problems. These misclassifications highlighted the need for further refinement of the API's error-detection algorithms to enhance accuracy.

It is important to notice that the feedback mechanism, while improved, still requires further development to ensure that error messages are not only instructional but also tailored to different learning levels and contexts. This personalization of feedback could be achieved through adaptive learning technologies that adjust the complexity and type of feedback based on the student's proficiency level and learning progression.

Future improvements could also focus on expanding the range of mathematical operations and concepts covered by the API. Currently, the API handles basic operations such as addition, subtraction, multiplication, and division, but incorporating more advanced topics such as algebra, geometry, and calculus would significantly broaden its educational impact. Furthermore, integrating natural language processing capabilities could help in understanding and classifying errors in word problems, which are often more challenging for students.

5. Conclusion

Development and evaluation of this MSEC API is a step towards advancement in educational technology that suits environments with limited resources. There is a need for providing quick, precise feedback on mathematical problem-solving, which goes a long way to bring out the learning process effectively so that analytical and critical thinking skills are developed. By automatically classifying errors like syntactic mistakes, conceptual misunderstanding, calculation mistakes, etc., it provides a solution befitting ITS-U.

The practical application of MSEC was demonstrated in a single case including 49 students. The iterative design processes under the guidance of DSR should ensure that MSEC is theoretically valid and practically feasible. Valuable educator feedback refined the API to be more user-friendly and effective. However, the study also found some limitations, which include dealing with more complex mathematical problems and the provision of more personalized feedback. Future work will focus on expanding MSEC's capabilities to cover a broader range of mathematical concepts and incorporating adaptive learning technologies for further personalization. The MSEC API is a tool that enhances educational practices by helping educators deliver personalized instruction and improve student learning outcomes through immediate, targeted feedback.

6. Artifacts Availability

The artifacts generated from this study are available from the corresponding author.

Acknowledgments

This project was funded by the Ministry of Education - MEC (TED 11476). We acknowledge the use of generative artificial intelligence tools, such as ChatGPT (3.5), Grammarly, and Google Translate, to aid in writing and revising this paper. The authors conducted a thorough review of the text and assume full responsibility for its content.

References

Brasil. Ministério da Educação (2018). Base nacional comum curricular. Brasília: MEC.

Conati, C. and Merten, C. (2014). Impact of adaptive feedback on learning and behavior. *User Modeling and User-Adapted Interaction*, 24(5):413–431.

Davis, S. R., DeCapito, C., Nelson, E., Sharma, K., and Hand, E. M. (2020). Homework helper: Providing valuable feedback on math mistakes. In *Advances in Visual Computing: 15th International Symposium, ISVC 2020, San Diego, CA, USA, October 5–7, 2020, Proceedings, Part II 15*, pages 533–544. Springer.

Gasevic, D., Paul, P., Chen, B., Fan, Y., Rodrigo, M. M., Cobo, C., and Cecilia, A. (2018). *Learning Analytics for the Global South*. Foundation for Information Technology Education and Development, Quezon City, Philippines, published edition.

Hattie, J. and Timperley, H. (2007). The power of feedback. *Review of Educational Research*, 77(1):81–112.

Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS quarterly*, pages 75–105.

Hillmayr, D., Ziernwald, L., Reinhold, F., Hofer, S. I., and Reiss, K. M. (2020). The potential of digital tools to enhance mathematics and science learning in secondary schools: A context-specific meta-analysis. *Computers & Education*, 153:103897.

Isotani, S., Bittencourt, I. I., Challco, G. C., Dermeval, D., and Mello, R. F. (2023). Aied unplugged: Leapfrogging the digital divide to reach the underserved. In Wang, N., Rebolledo-Mendez, G., Dimitrova, V., Matsuda, N., and Santos, O. C., editors, *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners, Doctoral Consortium and Blue Sky*, pages 772–779, Cham. Springer Nature Switzerland.

Levin, T. and Levin, I. (2012). Fostering mathematical thinking through cognitive scaffolding. *Educational Psychology Review*, 24(1):401–418.

Lin, C.-C., Huang, A. Y., and Lu, O. H. (2023). Artificial intelligence in intelligent tutoring systems toward sustainable education: a systematic review. *Smart Learning Environments*, 10(1):41.

Mousavinasab, E., Zarifsanaiey, N., R. Niakan Kalhori, S., Rakhshan, M., Keikha, L., and Ghazi Saeedi, M. (2021). Intelligent tutoring systems: a systematic review of characteristics, applications, and evaluation methods. *Interactive Learning Environments*, 29(1):142–163.

Nkambou, R., Mizoguchi, R., and Bourdeau, J. (2010). *Advances in intelligent tutoring systems*, volume 308. Springer Science & Business Media.

Patel, N., Thakkar, M., Rabadiya, B., Patel, D., Malvi, S., Sharma, A., and Lomas, D. (2022). Equitable access to intelligent tutoring systems through paper-digital integration. In *Intelligent Tutoring Systems: 18th International Conference, ITS 2022, Bucharest, Romania, June 29–July 1, 2022, Proceedings*, pages 255–263. Springer.

Polya, G. (2014). *How to Solve It: A New Aspect of Mathematical Method*. Princeton University Press.

Rodrigues, L., Guerino, G., Silva, T. E., Challco, G. C., Oliveira, L., da Penha, R. S., Melo, R. F., Vieira, T., Marinho, M., Macario, V., et al. (2024). Mathaide: A qualitative study of teachers' perceptions of an its unplugged for underserved regions. *International Journal of Artificial Intelligence in Education*, pages 1–29.

Rodrigues, L., Pereira, F. D., Marinho, M., Macario, V., Bittencourt, I. I., Isotani, S., Dermeval, D., and Mello, R. (2023). Mathematics intelligent tutoring systems with handwritten input: a scoping review. *Education and Information Technologies*, pages 1–27.

Santos, F. et al. (2018). *National Curriculum Standards of Brazil: Challenges and Perspectives*. Springer.

Shute, V. J. (2008). Focus on formative feedback. *Review of Educational Research*, 78(1):153–189.

Steenbergen-Hu, S. and Cooper, H. (2014). A meta-analysis of the effectiveness of intelligent tutoring systems on college students' academic learning. *Journal of educational psychology*, 106(2):331.

VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education*, 16(3):227–265.

Veloso, T. E., Chalco Challco, G., Rogrigues, L., Versuti, F. M., Sena da Penha, R., Silva Oliveira, L., Corredato Guerino, G., Cavalcanti de Amorim, L. F., Monteiro Marinho, M. L., Macario, V., Dermeval, D., Bittencourt, I. I., and Isotani, S. (2023). Its unplugged: Leapfrogging the digital divide for teaching numeracy skills in underserved populations. In *Towards the Future of AI-augmented Human Tutoring in Math Learning 2023 - Proceedings of the Workshop on International Conference of Artificial Intelligence in Education co-located with The 24th International Conference on Artificial Intelligence in Education (AIED 2023)*. Springer.

Vinuesa, R., Azizpour, H., Leite, I., Balaam, M., Dignum, V., Domisch, S., Felländer, A., Langhans, S. D., Tegmark, M., and Fuso Nerini, F. (2020). The role of artificial intelligence in achieving the sustainable development goals. *Nature communications*, 11(1):233.