CZU: 303.622 **https://doi.org/10.59295/sum5(195)2026_20**

METHODOLOGY FOR CREATING GRADING RUBRICS FOR OPEN-ENDED QUESTIONS

Mihail CROITOR,
Moldova State University

Assessing open-ended questions requires explicit criteria and transparent procedures, especially in automated assessment contexts. This paper proposes a generalized framework, with mathematical formalization, for designing grading rubrics suitable for AI-based evaluation systems. Four rubric types are systematized: binary rubrics, rubrics with multiple compliance levels, rubrics with penalty points, and rubrics with weighting coefficients, including combined variants. For each type, formal equations for final-grade computation are provided, supporting algorithmic implementation. A seven-step rubric-development methodology is presented, emphasizing criterion completeness, formulation clarity, operationalization, and structural consistency. The proposed framework is platform-independent and illustrated through examples from programming education. It also supports comparability across rubric designs and assessment contexts.

Keywords: *grading rubrics, open-ended questions, automated assessment, criterion-based assessment, learning objectives, educational measurement, formative assessment.*

METODOLOGIA CREĂRII RUBRICILOR PENTRU EVALUAREA ÎNTREBĂRILOR DE TIP DESCHIS

Evaluarea întrebărilor deschise necesită criterii explicite și proceduri transparente, în special în contextul evaluării automatizate. Lucrarea propune un cadru generalizat, cu formalizare matematică, pentru proiectarea rubricilor de evaluare aplicabile în sisteme bazate pe inteligență artificială. Sunt sistematizate patru tipuri de rubrici: binare, cu niveluri multiple de conformitate, cu penalizări și cu coeficienți de ponderare, inclusiv variante combinate. Pentru fiecare tip sunt prezentate formule de calcul al notei finale, adecvate implementării algoritmice. Este descrisă o metodologie în șapte pași pentru elaborarea rubricilor, cu accent pe completitudinea criteriilor, claritatea formulărilor, operaționalizare și structurare. Cadrul propus este independent de platformă și este ilustrat prin exemple din domeniul programării.

Cuvinte-cheie: *rubrici de evaluare, întrebări deschise, evaluare automatizată, evaluare criterială, obiective de învățare, măsurare educațională, evaluare formativă.*

Introduction

Assessing student knowledge is an important part of the educational process. It allows educators to objectively evaluate students' understanding of the material and their skills, and to provide constructive feedback.

Knowledge assessment can be performed using various methods, referred to in this paper as **assessment tools**: individual assignments, reports, essays, tests, among others [1]. An assessment tool is any method or instrument used to measure and evaluate students' knowledge, skills, and competencies, consisting of the following main elements:

1. the name of the assessment tool;
2. the learning objectives that the assessment tool checks;
3. the formulation of the task or tasks;
4. the assessment criteria.

An assessment tool may also include additional elements: *description, instructions for students, examples of tasks, and assessment recommendations*. Assessment tools include: *question, test, individual assignment, group project*.

The simplest (atomic) assessment tool is a question. Other assessment tools consist of one question or a set of questions.

Questions can be classified by the type of response required from the student. There are two main types of questions:

- closed-ended questions;
- single-choice questions;
- multiple-choice questions;
- short-answer questions;
- matching questions;
- ordering questions; etc.
- open-ended questions;
- essay questions;
- math questions;
- report questions;
- programming questions; etc.

Closed-ended questions require the student to select one or more correct answers from a given set of options. This type of question is well suited to automated grading due to the presence of a clearly defined correct answer and is often used to assess basic knowledge and facts.

Open-ended questions require the student to provide an extended response involving analysis, synthesis, and evaluation of information. The student must demonstrate their knowledge and skills in a free-form manner. In the absence of a clearly defined correct answer, assessing the response may be unclear to students or perceived as subjective. In such cases, it is recommended to use rubrics for assessing student responses, as this ensures transparency and objectivity in the assessment process [13, 17].

The problem of assessing open-ended questions is especially relevant in the context of large-scale online education, where student cohorts can number in the thousands and instructional resources are limited for high-quality assessment of each response. Automated Essay Scoring (AES) is an actively developing research area [11, 20]. Traditional AES systems were based on machine learning and natural language processing [5, 8], but in recent years significant progress has been made thanks to the application of modern artificial intelligence methods, including large language models [10, 18].

Recent studies demonstrate the high potential of using machine learning and artificial intelligence methods for automated assessment of students' textual responses [4, 7]. For example, Mizumoto and Eguchi [10] showed the effectiveness of using AI models for essay assessment, and Uyar and Büyükahiska [18] explored the capabilities of modern language models as a tool for automated assessment. Hashemi et al. [7] proposed a multidimensional calibrated approach to automated assessment, emphasizing the importance of clearly structured rubrics for the effective operation of AI-based systems.

However, despite recent progress in the application of artificial intelligence for assessment, the question of a systematic representation of grading rubrics suitable for automated assessment remains open. Rubrics for automated systems must be sufficiently clear, unambiguous, and structured so that algorithms can correctly interpret the criteria and apply them consistently to student responses. Rubrics designed for manual grading do not always meet these requirements [7, 9]. As noted in the literature, rubric development is an iterative and complex process with both pedagogical and technical components [1, 3, 6, 17].

Various learning management systems (such as Moodle) implement mathematical models of assessment using rubrics with multiple levels of correspondence. However, these models are designed for specific platforms and are not presented in a generalized form suitable for theoretical analysis and adaptation to various contexts of automated assessment.

The scope of this work is to develop a generalized representation of analytic grading rubrics for open-ended questions, with a formalized mathematical description suitable for automated assessment.

To achieve this goal, the following **tasks** need to be addressed:

1. Systematize the principles of criterion-based assessment and their connection to learning objectives.
2. Develop a generalized representation of the main types and structures of grading rubrics for open-ended questions.

3. Provide mathematical formalization of different rubric variations (multi-level rubrics, penalties, weighting coefficients).

4. Describe the process of formulating assessment criteria based on learning objectives.

5. Develop a methodology for creating rubrics that takes into account the requirements of clarity and unambiguity for automated assessment systems.

The scientific novelty of this work is as follows:

1. A generalized representation of analytic grading rubrics is proposed, combining different assessment approaches (binary, multi-level, with penalties, with weighting coefficients) into a unified conceptual model.

2. A mathematical description of various types of rubrics has been developed, allowing formalization of the assessment process and ensuring its technical feasibility in automated systems.

3. An explicit connection between learning objectives, assessment criteria, and rubric structure has been established, providing both pedagogical justification and technical applicability of rubrics.

4. A methodology for creating rubrics has been proposed, taking into account the requirements for clarity and unambiguity of the assessment results.

The article is organized as follows. The section “Criterion-based Assessment” discusses the connection between learning objectives and assessment criteria. The section “Grading Rubrics” describes the main rubric types and structures. The sections “Variations of Grading Rubrics” and “Methodology for Creating Grading Rubrics” present the development process for different types of tasks. The final part formulates the main conclusions and directions for further research.

1. Criterion-Based Assessment

When designing a course, learning objectives are defined that students should achieve upon completion of the course. These objectives may include knowledge, skills, and competencies that students should acquire. Each course topic corresponds to the set objectives to some extent and is aimed at achieving them. Course topics implement more specific learning tasks, which collectively determine the achievement of learning objectives. Additionally, learning tasks can be broken down into smaller session objectives, which define specific learning outcomes for each session.

For effective assessment of students’ knowledge, it is necessary to clearly define a method for measuring the correspondence of their knowledge and skills to the course objectives. An optimal approach is criterion-based assessment, where each criterion corresponds to a specific learning objective [1, 3, 16].

The alignment of a student’s knowledge and skills with the course objectives can be naturally expressed as a percentage:

- 100% – full correspondence of knowledge and skills to the course objectives;
- 0% – complete non-correspondence;
- 1%–99% – intermediate levels of correspondence.

This percentage can be mapped to any conventional grading scale.

Example: Python Programming for Beginners

Consider a course “Python Programming for Beginners” aimed at teaching students the basics of programming using the Python language. The course may have the following general learning objectives:

- Understand the basic concepts of programming, such as variables, data types, operators, and control structures (forming basic programming knowledge);
- Develop simple programs in Python using functions and modules (forming programming skills);
- Solve basic programming problems using algorithms and data structures (forming analytical skills);
- Write and debug Python code using development tools (forming technical skills).

In this case, the first five course topics may be formulated as follows:

1. Introduction to Programming and Python
2. Variables and Data Types
3. Operators and Expressions
4. Control Structures (Conditional Statements and Loops)
5. Functions and Modules

The first course topic “Introduction to Programming and Python” is aimed at achieving the general

learning objective related to understanding the basic concepts of programming and includes the following learning tasks:

- Familiarization with the history and features of the Python language:

- when and by whom the Python language was created;
- where Python is used;
- the main advantages of Python;
- how Python differs from other programming languages.

- Installation and configuration of the development environment for Python:

- installation of Python on various operating systems;
- selection and installation of an integrated development environment (IDE) for Python;
- configuration of the development environment for effective work with Python.

- Familiarization with the basic syntax of Python:

- basic Python syntax for creating and running programs;
- using comments in Python code;
- basic rules for formatting code in Python.
- running a program and checking its operation.

All the learning objectives together determine the achievement of the general learning objectives of the course.

Specific learning objectives (lesson objectives, learning tasks) form the basis for developing assessment criteria. Each objective formulated for a lesson or topic must be measurable and verifiable, thus becoming a natural criterion for a grading rubric.

2. Grading Rubrics

In educational practice and learning management systems, various approaches are used to create and apply grading rubrics. For example, the Moodle system provides functionality for creating rubrics with multiple levels and automatic calculation of grades based on them. Similar mechanisms are implemented in other LMS (Learning Management Systems). However, these implementations are platform-specific and are not presented as a generalized model that could serve as a theoretical basis for the development and comparison of different types of rubrics.

Existing scientific works consider separate aspects of rubrics: elements of rubrics [2], validation for specific subject areas [15], comparison of analytical and holistic approaches [19]. However, there is no unified generalized representation that systematizes various types of rubrics and their mathematical description.

As already noted, various assessment tools are used to evaluate students' knowledge, which define assessment criteria. These criteria can be formulated in the form of a grading rubric.

According to Reneau [13], **a grading rubric is a tool that articulates the expectations for an assignment by listing criteria, and for each criterion, describing levels of quality. It serves as a guide for grading, consisting of specific predetermined performance criteria used in evaluating students' work**. A grading rubric is a matrix where rows correspond to assessment criteria, and columns correspond to levels of compliance with these criteria. This matrix allows for systematic evaluation of a student's work for each criterion and determining the final grade based on the cumulative compliance.

The formation of assessment criteria is based on specific learning objectives: each lesson objective can become a separate criterion for evaluating a student's work. The assessment tool may cover all or part of the objectives. Meeting the criterion is scored as 100%, and not meeting it as 0%. A **grading rubric** is a set of objectives verified by the assessment tool, indicating how to determine full or partial compliance with these objectives. The simplest grading rubric has the form:

Table 1. The simplest grading rubric

Criterion	Complete Non-Compliance	Complete Compliance
Objective 1	0	100
Objective 2	0	100
...		
Objective n	0	100

The final grade for a student's response is determined as the sum of the percentages for each criterion divided by the number of criteria:

$$G = \frac{\sum_{i=1}^{N} P_i}{N} \quad (1)$$

Where:

G – the final grade for the student's response (in percentage);

N – the number of criteria in the grading rubric;

Pi – the percentage of the student's response compliance with the i-th criterion (0% or 100% in the simplest grading rubric).

Example: The simplest grading rubric for "Python Programming for Beginners"

An example assessment tool could be an assignment to write a small and simple Python program for the first topic of the course "Introduction to Programming and Python". The grading rubric for this assignment is formed based on the lesson objectives and may look as follows (Table 2):

Table 2. The simplest grading rubric for Python programming

Criterion (based on lesson objectives)	Complete Non-Compliance	Complete Compliance
Basic Python syntax	0	100
Use of comments in Python code	0	100
Python code formatting rules	0	100
Running the program and checking its operation	0	100

An example student response is shown in Listing 1:

Listing 1. Example student response

```
# This is a simple Python program. The code contains an error - echo commands do not exist in Python.
echo("Hello, World!")
```

In this case, the grading according to the rubric will be as follows (Table. 3):

Table 3. Student response grading according to the simplest rubric

Criterion	Complete Non-Compliance	Complete Compliance
Basic Python syntax	0	100
Use of comments in Python code	0	100
Python code formatting rules	0	100
Running the program and checking its operation	0	100

The final grade for the student's response will be:

$$G = \frac{0\%+100\%+100\%+0\%}{4} = 50\% \quad (2)$$

This example illustrates that the rubric criteria directly correspond to the specific learning objectives for the first topic described above. Each objective becomes a measurable criterion by which the student's work is evaluated [12, 15].

3. Variations of Grading Rubrics

Simple rubrics are often insufficient in practice because responses can be partially correct or may contain errors of different severity. For such situations, extended rubrics are useful: they preserve the same basic logic but introduce intermediate compliance levels, penalties, and/or weights [2, 6, 19].

3.1. Multi-level Compliance

Instead of a binary scheme (0%/100%), criteria can be evaluated on multiple levels. A 5-level scale can be used: 0%, 25%, 50%, 75%, 100%, where each level corresponds to a certain response quality *(Table 4)*.

Table 4. Compact description of compliance levels

Percentage	Interpretation	Meaning
0%	Completely incorrect	The response does not meet the criterion
25%	Almost incorrect	The response contains significant errors
50%	Partially correct	Partial correspondence with the criterion
75%	Almost correct	Nearly complete correspondence
100%	Correct	Complete correspondence

The final grade is calculated in the same way as in the simple case:

$$G = \frac{\sum_{i=1}^{N} P_i}{N} \quad (3)$$

where P_i is the compliance level assigned to criterion i and N is the total number of criteria.

However, each criterion may define its own levels and interpretations, which allows the rubric to be adapted to the task specifics and learning objectives.

Using multi-level criteria reflects response quality more precisely and provides more differentiated feedback, supporting competence development and reinforcing understanding [9, 17].

3.2. Penalties

Penalties can be applied per criterion or globally. For per-criterion penalties:

$$G = \frac{\sum_{i=1}^{N} max(0,100-k_i \cdot D_i)}{N} \quad (4)$$

where $\boldsymbol{k_i}$ is the number of errors for criterion ***i*** and $\boldsymbol{D_i}$ is the penalty per error.

For global penalties:

$$G = max\left(0, \frac{\sum_{i=1}^{N} P_i}{N} - k \cdot D\right) \quad (5)$$

where k is the total number of errors and D is the penalty per error.

Penalties are useful to account for errors that may be present even in formally correct responses.

3.3. Weighting coefficients

When criteria have different importance, weights ***qi*** are used:

$$G = \frac{\sum_{i=1}^{N} P_i \cdot q_i}{\sum_{i=1}^{N} q_i} \quad (6)$$

Weighting prioritizes criteria with greater impact on the final result.

3.4. Combined rubrics (unified example)

In practice, mechanisms can be combined: compliance levels, penalties, and weights. For each criterion, the final score can be computed as:

$$P_i = max(0, L_i - k_i \cdot D_i) \quad (7)$$

where $L_i \in \{0,25,50,75,100\}$ is the compliance level, k_i is the number of errors, and D_i is the penalty for one error.

The final grade is determined as the weighted average of criterion scores:

$$G = \frac{\sum_{i=1}^{N} P_i \cdot q_i}{\sum_{i=1}^{N} q_i} \quad (8)$$

Table 5. Combined rubric (Python example)

Criterion	0%	25%	50%	75%	100%	Errors	Penalty	Weight	Rezult
Basic Python syntax				■		**1**	**5**	**3**	**70**
Use of comments in Python code					■	**0**	**5**	**1**	**100**
Python code formatting rules					■	**0**	**5**	**1**	**100**
Running the program and checking its work			■			**2**	**10**	**2**	**30**
Total									**67,14**

The final result for the unified example is:

$$G = \frac{70\cdot3+100\cdot1+100\cdot1+30\cdot2}{3+1+1+2} = \frac{470}{7} \approx 67{,}14 \quad (9)$$

The general formula for combined rubrics is:

$$G = \frac{\sum_{i=1}^{N} max(0, L_i - k_i \cdot D_i)\cdot q_i}{\sum_{i=1}^{N} q_i} - k \cdot D \quad (10)$$

– G the final grade for the student's response;
– N the number of criteria;
– L_i the compliance level for criterion i;
– k_i the number of errors for criterion i;
– D_i the penalty per error for criterion i;
– q_i the weight of criterion i;
– k the total number of errors;
– D the global penalty per error.

Typically, only a subset of these mechanisms is applied depending on the task specifics and learning objectives (e.g., D can be zero if global penalties are not used).

4. Methodology for Creating Grading Rubrics

The rubric types and their connection to learning objectives were analyzed above. Based on this analysis, a rubric-development methodology for open-ended questions can be formulated. To create an effective rubric, the following seven-step sequence is recommended:

1. define the learning objectives verified by the assessment tool;
2. identify measurable criteria;
3. define compliance levels (e.g., 0%, 25%, 50%, 75%, 100%) with operational descriptions rather than vague labels;
4. assign weighting coefficients (if needed);
5. define penalty points (if needed);
6. test the rubric on real student work to verify objectivity [14];
7. adjust the rubric based on results (returning to step 3 if necessary).

Characteristics of Rubrics for Automated Assessment

For use in automated assessment systems, a rubric should satisfy the following requirements:

1. **Completeness of criteria** – all aspects to be assessed must be clearly defined as separate criteria.
2. **Clarity of formulations** – each compliance level should describe observable characteristics of the response, eliminating ambiguity of interpretation.
3. **Operationalization of criteria** – criteria should be expressed via measurable or verifiable indicators.
4. **Avoidance of subjective assessments** – formulations like “qualitative analysis” or “deep understanding” must be concretized through observable indicators.
5. **Structuredness** – the rubric should have a clear structure enabling formal representation (e.g., as tables, JSON, XML).

These conditions enable programmatic implementation and increase assessment reproducibility.

Results and Discussion

The proposed generalized model covers the main types of analytic grading rubrics used in practice and provides a single mathematical basis adaptable to different automated-assessment contexts.

Formalizing analytic grading rubrics (binary, level-based, with penalties, with weights, combined) enables:
formalization of the assessment process;
transparency and reproducibility of grades;
easier technical implementation in automated systems;
comparison of assessment approaches on a common basis.

Limitations: The methodology focuses on quantitative criterion-based assessment (analytic rubrics). It does not cover holistic assessment [19] and does not detail calibration/validation of rubrics in specific systems.

Conclusions

This work presents a generalized description of grading rubrics for open-ended questions, oriented toward use in automated grading systems.

In accordance with the stated objectives, the following results were obtained:

1. The principles of criterion-based assessment have been systematized, showing an explicit connection between learning objectives (course, topic, lesson) and grading criteria in rubrics.

2. A generalized representation of grading rubrics has been developed, including four main types: binary rubrics, rubrics with multiple compliance levels, rubrics with penalty points, and rubrics with weight coefficients, as well as their combinations.

3. A mathematical description for each type of rubric has been proposed, allowing formalization of the final grade calculation process and ensuring technical feasibility in automated systems.

4. The process of formulating grading criteria based on specific learning objectives has been described, wherein each measurable objective becomes a separate rubric criterion.

5. A seven-step methodology for creating rubrics has been developed, indicating specific requirements for automated assessment (completeness, unambiguity, operationalization of criteria).

The proposed representation is independent of platform and technology, which distinguishes it from local, platform-specific solutions. Mathematical formalization increases assessment transparency and simplifies technical implementation [4]. Clear structure and criterion unambiguity make rubrics suitable both for manual grading and for AI-based automated assessment systems [1, 13, 17].

Promising directions for future research include empirical validation of the proposed rubric types, development of methods for automatic generation of criteria, and investigation of the consistency of automated assessment with expert assessments by instructors.

Bibliography:

1. ALLEN, D., TANNER, K. Rubrics: Tools for Making Learning Goals and Evaluation Criteria Explicit for Both Teachers and Learners. *CBE—Life Sciences Education*, 2006, vol. 5, nr. 3, pp. 197-203.
2. AUSTRALIAN NATIONAL UNIVERSITY (ANU). *Elements of rubrics* [online]. Canberra: ANU Learning and Teaching, 2024 [viewed 18 December 2025]. Available from: https://www.anu.edu.au/students/academic-skills/learning-and-teaching/elements-of-rubrics
3. BARRIE, S., BREW, A., MCCULLOCH, M. Qualitatively Different Conceptions of Criteria Used to Assess Student Learning. Communication is presented AARE Annual Conference, Melbourne, November 1999.
4. BULUT, O. et al. The Rise of Artificial Intelligence in Educational Measurement: Opportunities and Ethical Challenges. *Computers and Education: Artificial Intelligence*, 2024, vol. 6, articolul 100214. DOI: https://doi.org/10.1016/j.caeai.2024.100214
5. BURSTEIN, J., LEACOCK, C., SWARTZ, R. *Automated Evaluation of Essays and Short Answers* [online] 2019. Figshare. [viewed 31 October 2024]. Available from: https://hdl.handle.net/2134/1790

6. DOUGHTY, L., CABALLERO, M.D. Rubric Design For Separating The Roles Of Open-Ended Assessments. În: *Physics Education Research Conference Proceedings*. Minneapolis, MN, 2014, pp. 71-74.
7. HASHEMI, H., EISNER, J., ROSSET, C., VAN DURME, B., KEDZIE, Ch. LLM-RUBRIC: A Multidimensional, Calibrated Approach to Automated Evaluation of Natural Language Texts. *arXiv preprint* [online] 2024. arXiv:2410.00640. [viewed 18 December 2025]. Available from: https://arxiv.org/abs/2410.00640
8. KUMAR, R., MATHIAS, S., SAHA, S., BHATTACHARYYA, P. Many Hands Make Light Work: Using Essay Traits to Automatically Score Essays. În: *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, vol.35, nr.15, pp.13367-13374.
9. MCKENNA, M., DEDRICK, R.F., GOLDSTEIN, H. Development and Initial Validation of the Early Elementary Writing Rubric to Inform Instruction for Kindergarten and First-Grade Students. *Assessment for Effective Intervention*, 2022, vol.47, nr.4, pp.220-233. DOI: https://doi.org/10.1177/15345084211065977
10. MIZUMOTO, A., EGUCHI, M. Exploring the Potential of Using an AI Language Model for Automated Essay Scoring. *Research Methods in Applied Linguistics*, 2023, vol.2, nr.2, article 100050. DOI: https://doi.org/10.1016/j.rmal.2023.100050
11. RAMESH, D., SANAMPUDI, S.K. An automated essay scoring systems: a systematic literature review. *Artificial Intelligence Review*, 2021, vol.55, nr.3, pp.2495-2527. DOI: https://doi.org/10.1007/s10462-021-10068-2
12. RAHMAN, Md. M., WATANOBE, Yu., SHIRAFUJI, A., HAMADA, M. Exploring Automated Code Evaluation Systems and Resources for Code Analysis: A Comprehensive Survey. *Journal of Information Processing*, 2018, vol. 26, pp. 471-486.
13. RENEAU, F.. *Rubric Development 101* [online]. Office of Academic Effectiveness, Georgia Tech. [viewed 18 December 2025]. Available from: https://oae.gatech.edu/rubric-development-101
14. RIMON, Amos, 2025. Improving Inter-Rater Reliability: Best Practices and Strategies. *Deepchecks Blog* [online]. [viewed 18 December 2025]. Available from: https://deepchecks.com/blog/
15. SAITO, Daisuke, YAJIMA, Risei, WASHIZAKI, Hironori and FUKAZAWA, Yoshiaki, 2021. Validation of Rubric Evaluation for Programming Education. *Education Sciences*, vol. 11, nr. 10, articolul 656. DOI: https://doi.org/10.3390/educsci11100656
16. ŞOVA, Tatiana and PUTINĂ, Dorina, 2017. *Evaluarea în învăţământ: Suport de curs*. Bălţi: Universitatea de Stat „Alecu Russo» din Bălţi. ISBN 978-9975-3184-0-2.
17. TRACTENBERG, Rochelle E., 2021. The Assessment Evaluation Rubric: Promoting Learning and Learner-Centered Teaching through Assessment in Face-to-Face or Distanced Higher Education. *Education Sciences*, vol. 11, nr. 8, articolul 441. DOI: https://doi.org/10.3390/educsci11080441
18. UYAR, Ahmet Can and BÜYÜKAHISKA, Dilek, 2025. Artificial intelligence as an automated essay scoring tool: A focus on ChatGPT. *International Journal of Assessment Tools in Education*, vol. 12, nr. 1, pp. 20–32. DOI: https://doi.org/10.21449/ijate.1517994
19. YEO, Siew Wan, SIGNORELLI, Christina, VO, Khanh, SMITH, Greg, 2024. A Retrospective Cohort Analysis Comparing Analytic and Holistic Marking Rubrics in Medical Research Education. *Journal of Medical Education and Curricular Development*. DOI: https://doi.org/10.1177/23821205241277337
20. ZUPANC, Kaja, BOSNIĆ, Zoran, 2015. Advances in the Field of Automated Essay Evaluation. *Informatica*, vol. 39, pp. 383–395.

Data about author:
Mihail CROITOR, University Lecturer, Moldova State University, Chisinau, Republic of Moldova.
ORCID: 0009-0004-5046-0811
E-mail: mihail.croitor@usm.md

Presented: 27.02.2026
Reviewed: 30.04.2026
Accepted for publication: 20.05.2026