The Determination of the Learning Performance based on Assessment Item Analysis

Doru Anastasiu Popescu[1,*,†], Ovidiu Domșa[2,†] and Nicolae Bold[3,†]

[1]Department of Mathematics and Computer Science, University of Pitești, Romania
[2]1 Decembrie 1918 University of Alba Iulia, Romania
[3]UASVM Bucharest, Faculty of Management and Rural Development, Slatina Branch, Slatina, Romania

Abstract

The analysis of the performance of the educational process is one of the essential aspects of the contemporary approach of the educational system. Technology has permitted the analysis of various components of the learning process, which has developed in the process of learning analytics. This paper presents the model and implementation of a concept that uses learning analytics to determine the outcome of an educational process and its performance. Its performance refers to the group understanding of a specific concept measured using the results to systematic evaluation during a period of time. The model, called Course Item Management Generation (CIM-GET), is part of a larger model that is centered on the educational assessment process and which uses machine learning-based techniques and evolutionary algorithms to generate assessment tests used for learning purposes. The current model uses statistical and item response analysis parameters in order to create a report regarding the items within the tests that are given over a period of time to specific students within a faculty of university.

In the first part, the CIM-GET model will be presented in the context of the larger model called Dynamic Model for Assessment and Interpretation of Results (DMAIR), then several results obtained after the technical and statistical implementation will be presented. The CIM-GET model uses items from an item dataset, extracted using machine learning-based tools by the defining keywords of the item, which also represent the topics of an item, which form an optimal test using a generation algorithm (e.g., genetic algorithm). After the test is given to students, the results are stored in a database, a report is output and a list of topics that need to be revised is generated. In this matter, the practical results of the presented model will be shown, in order to show the practical importance of the results.

Keywords

assessment, education, item analysis, test, answer evaluation

1. Introduction

The development of the computing models and method has permitted the creation of several research fields that study and implement these models and methods in the educational domain. In this matter, the social context and the problems that arose regarding the educational activity were also a catalyst in order for this phenomenon to occur. As a result, the activity of a typical

WSDM 2023 Crowd Science Workshop on Collaboration of Humans and Learning Algorithms for Data Labeling, March 3, 2023, Singapore

*Corresponding author.
†These authors contributed equally.
dopopan@gmail.com (D. A. Popescu); domsa.ovidiu@gmail.com (O. Domșa); bold1_nicolae@yahoo.com (N. Bold)

teacher has been changing due to the inclusion of the technological developments, especially in educational management.

We should also take into consideration the major subject of recent research in the educational domain, which is based on creating meaningful and effective educational process, especially using digital technology and computing-based methods. The objective of an educational process is the fulfillment of the objectives and one of the main possibilities to achieve these objectives is related to an objective assessment. The objectivity of an assessment process is related to the appropriate design of the assessment tools and the valid analysis of the assessment results, which can be conceptually accomplished by the usage of specific pedagogical methods, such as Universal Design for Learning (UDL) [1] or, more specifically, Universal Design for Assessment (UDA) [2], technically implemented using computing methods, such as machine learning, evolutionary algorithms and thoroughly studied using, for example, Learning Analytics (LA) [3] or statistical indicators [4, 5]. The main objective of a successful automated design and analysis of an assessment test is to be as close as a human-centered approach of a design result with similar requirements, because human experience is still hard to be surpassed in terms of the specific assessment test and item design and analysis [6].

In order to achieve such a specific objective, any type of assessment design must take into consideration design and analysis frameworks that check four major aspects [7]: communication, orientation, learning experience and evaluation, with regard to reliability and validity of the assessment. While communication refers to the correct reciprocal transmission and understanding of the assessment objectives to all the participants to the assessment, the orientation refers to the optimal choice of the assessment form based on the studied content. As for the other two aspects, the learning experience takes into account the closeness of the assessment to the real-life situations and the validity stands for the extent to which the assessment objectives were accomplished. Also, several factors of the assessment must be taken into consideration, such as subject content, electronic flexibility, language usage, format options, time limits or a direct link with the goals and objectives of the course.

This paper presents the development of the CIM-GET model, describing the architecture of the model, the implementation and its results. In short, the model consists in a specific method related to assessment item analysis in a specific educational context. In this matter, the model is based on the hypothesis that the statistical data related to an assessment item is influenced by the conceptual understanding of the item subject. Also, several factors, such as item-based factors (e.g., the degree of difficulty of the item, the theoretical / practical nature of the item, the item type, item number), statistical and item test factors (e.g., mean and standard deviation, item discrimination, item attempts, reliability coefficient), student-centered factors (e.g., student educational level) or group-centered factors (e.g., assessment score mean), which will be presented in the next sections, are to be taken into consideration regarding the item response. In this matter, given a specific period of educational time, such a semester or a year, and the periodic assessment of the students taken by a teacher, the item analysis conceptualized in the CIM-GET model, with its practical implementation, underlies the notions that can be elaborated more during the courses, due to lower rates of correct answers of the items that check these specific notions during the periodic assessment tests. The model is also enhanced by determining supplementary item verification mechanisms using automated methods of clustering of items in order to predict whether an item is prone to have lower rates of response

based on the factors taken into consideration, tendency that will be confirmed by the factual item analysis. Obviously, the automated clustering will provide finer results with a larger factor set taken into consideration.
The CIM-GET model is a modular part of an integrated model denoted by Dynamic Model for Assessment and Interpretation of Results (DMAIR), which is formed of three main components: the test generation, the answer check and the response analysis. For each component, a different model with its implementation will be described further. The DMAIR model takes into account several areas of educational assessment, especially regarding item generation using various methods, such as machine learning, natural language processing and evolutionary algorithms used to automatically generate items for a specific assessment test, with several requirements related to the test (the item subject, the degree of difficulty of the item, the theoretical / practical nature of the item and the item type). The CIM-GET model is used to analyze the answers, in the integrated model being responsible for the Answer Evaluation (AE) and Item Analysis (IA) parts.
For the detailed description of the model and its implementation, the paper is structured in several sections. In the first section, several literature landmarks and trends from the research fields are presented. The next section presents the description of the CIM-GET model and the integrated model that CIM-GET is part of, followed by a short description of a web-based implementation of the model and several results that show the practical potential of an implementation of this model.

2. Literature review

Extensive literature has been published regarding the optimization of the assessment process regarding design and analysis of the assessment components. In this matter, the most part of the automated educational assessment area consists in the development of assessment models and tools for Question Generation (QG) and Answer Evaluation (AE). An important part in the AE branch is dedicated to the Automated Essay Scoring (AES), as shown in [8, 9, 10], which has largely been researched in recent years.
Regarding QG branch, the majority of the research papers were directed to the generation of objective questions, such as multiple-choice [11, 12], true-false [13] or open-cloze questions [14, 15]. For a long time, classical subjects of QG research are related to the formulation of the questions from learning material, thus the recent research has been extensively related to sentence-to-question generation [16] and the generation of questions from any type of text [17], including artificial intelligence [18]. In order to visualize the extent of the research on this subject, an empiric research regarding article subjects in scientific databases revealed that the topic has a wide interest in the area of research. This research has been made based on a search operation on specific keywords (e.g., for the specific keyword „automatic question generation"). The search on the Google Scholar paper database returned 292 unique results for 2022. As for the methods used for the accomplishment of this task, one of the most used is the Natural Language Processing (NLP), which has been developed and refined over time.
For the AE branch, the research is focused on the short and essay answers analysis, which

also uses NLP-based techniques in order to accomplish the performant analysis of the text in the answer. One of the most researched topics is the evaluation of the correctness of the response, especially related to specific type of questions (e.g., multiple-choice questions [19, 20]). However, an increasing interest can be observed related to the automatic answer evaluation regarding essay-type items [21, 22].
Another important part of the research in educational assessment is related to Item Analysis (IA), a field situated at the border of several domains, such as statistics, psychometry, assessment or education. It showcases a wide range of research topics related to the mathematical and statistical aspects of assessment analysis [23], which remain landmarks regarding the item analysis topic and integrated intensively in learning management systems as basic functionalities for the human-centered analysis related to educational activity on a specific platform. The item analysis is an extremely important method on studying the student performance over given periods of time [24]. For this subjects, two approaches are thought to be the most fitted for item analysis: Classical Test Theory (CTT) and Item Response Theory (IRT). While CTT uses extensively statistical-based tools [25], such as proportions, averages and correlations, and is used for smaller-scale assessment contexts, the IRT has a more recent development and it is studied in respect to its more adaptive character [26]. The adaptive character of the IRT method consists in the important account given of the human factor related to the assessment process. One of the most important differences between the two approaches is based on the previous learning experience of the assessee, because IRT create an adaptive analysis based on a measurement precision which takes into account latent-attribute values, while CTT starts with the assumption that this precision is equal for all individuals [27]. In this paper, tangential concepts are used for the description of the development of CIM-GET model, especially regarding the statistical item analysis.
In a further development of the research literature, an important field which had recent serious practical implications in the educational process is the Deep Knowledge Tracking [28]. It has gained a lot of exposure in the recent period of the continuous development of online education, due to the fact that it proposes the analysis and prediction of the student educational behaviour based on previous personal learning experiences.
In order to accomplish the purpose of the current paper, we will use the literature concepts and follow a specific approach that has conceptual basis on several aspects of the cited literature. In this matter, the assessment item generation serves at a better integration of the assessment model and the IA cited literature shows an introductory part for the description of CIM-GET model.

3. Model description

3.1. DMAIR integrated model

The DMAIR model comprises several components that are essential to an assessment system. This system must be formed of three main functionalities: generation of items (func1 (I)), check mechanisms (func2 (II)) and answer evaluation (func3 (III)). While the module responsible for the generation of items, func1 (I), uses methods and tools for obtaining assessment tests suited to specific requirements, the check mechanism, func2 (II) is related to the validation of the answers

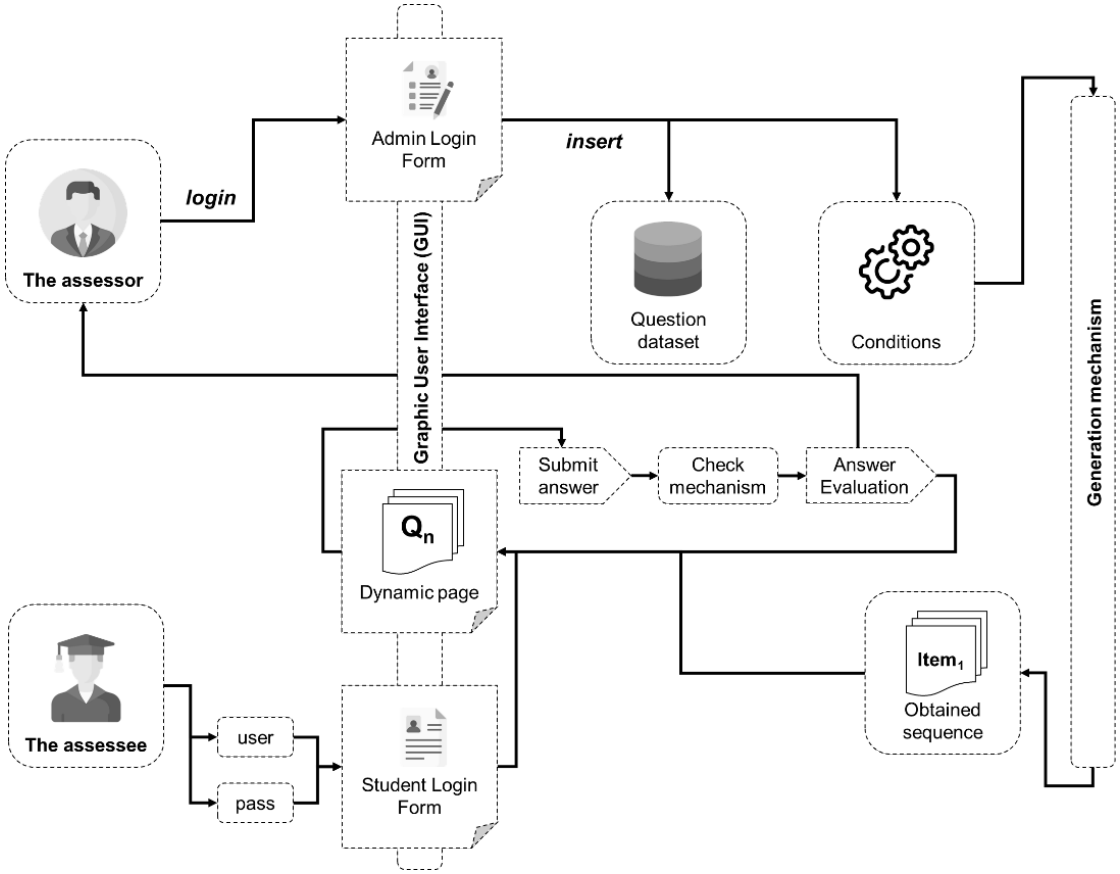

Figure 1: Visual depiction of the DMAIR model.

given by the users and the answer evaluation module, func3 (III), which will be presented further as the CIM-GET model, introduces the development of the item analysis for the given generated items and answers and is the most related to the learning analytics. A visual depiction of the model, including a graphical user interface component, is presented in the next figure.

3.1.1. Model structure

The main components of the model are the questions, named items after the generation process, the test, the requirements and the generation mechanism. The question is a particular case of an item, as well as a request or an exercise, this being the reason for which the questions will be considered particular cases of items and we will refer further to the questions, requests, exercises etc. as items. An item $q(id; st; dd; V; tp; t)$ is an object formed of the next components:

- the identification number of the item id_q, which has the role of being the unique identifier of the item in the implementation phase;
- the statement st_q, which is formed of a phrase or a set of phrases that describes the initial data and the requests of the item, which must be solved;

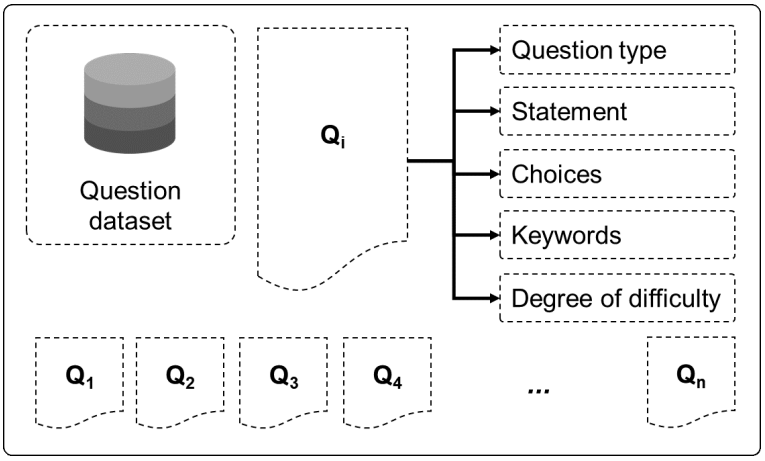

Figure 2: Visual representation of an item q.

- the set of keywords kw_q, which consists in the list of keywords that describe the best the topic of the item;
- the degree of difficulty dd_q, $dd\_q \in [0, 1]$; the degree of difficulty is calculated as the ratio between the number of incorrect answers at a specific item and the total number of answers. The degree of difficulty can also be calculated using the method presented in [29];
- choices set V_q (wherever necessary), which can be formed of a list of two or more possible answers when the item type is multiple or is null when the item type is short or essay;
- the theoretical or practical character of the question tp; $tp \in 0, 1$, where 0 is theoretical and 1 is practical;
- the item type t_q, $t\_q \in$ 'multiple', 'short', 'essay', illustrating the type of the item, whether it has choices or the answer is a textual one, given by the user, in case of short and essay types.

The item dataset, denoted further by $BD1$, contains items that are automatically generated using NLP methods or introduced manually by a teacher.

The $BD1$ dataset is schematically represented in Figure 2, where we can also see the main components of a general item: the item type (t_q), the statement st_q, the choice set V_q, the list of keywords kw_q and the degree of difficulty dd_q.
A test $T(S, DD, TP, QT)$ is a set of items q_i, $i = 1, S$, where S is the set of items that form the test and DD is the degree of difficulty of the test:

$$DD = \sum_{i=1}^{S} q_{dd_i} \tag{1}$$

In the equation, q_{dd_i} consists in the degree of difficulty of the item q_i within the test T.
The other components of a test are:

- TP is the theoretical-practical ratio, which gives the predominant type of the test. $TP \in [0, 1]$, the value of the ratio consisting in the proportion of the theoretical questions and the difference $1 \breve{} TP$ being the proportion of practical question;
- QT introduces the predominant item type in the test; QT is an array with three values: $[qt\_m, qt\_s, qt\_e]$. The values of the array contain the number of items of each type within the test, qt_m being the number of multiple-choice items, qt_s being the number of short-type items and qt_e being the number of essay-type items.

3.1.2. Model functionality

The generation mechanism uses a predefined set of actions that describe the generation and evaluation of the generated assessment tests.
Input data consists in:

- the desired subject given by the set of nr_k keywords generated by the user kw = kw_1, kw_2, ..., kw_{nr}_k;
- the number of questions required for each keyword nr_kw = nr_kw_1, nr_kw_2, ..., $nr\_kw_{nr}\_k$
- the desired degree of difficulty DD_u;
- the desired theoretical-practical ratio TP_u;
- the desired predominant question type QT_u = <qt_m, qt_s, qt_e>.

The model functionality contains the next main algorithms:

- the item generation algorithm (which will be denoted further by $GenTest$), correspondent to the functionality func1 (I), which contains actions related to the specific generation. This action can be established by following a specific set of steps:
  - **step 1**: Each keyword is parsed and for each of them a cluster of questions that have similar keywords with the current one is formed. The similarity is computed using NLP methods and the clusters are being formed using ML-based technique K-means. The nr_kw_i number of questions are taken into consideration for each kw_i keyword
  - **step 2**. The partial dataset of questions C_i that can be used for the generation of the test is formed. The main requirement taken into consideration is the subject of the test.
  - **step 3**. The test is generated based on other requirements using a specific type of method (e.g., genetic algorithms).
- the check mechanisms algorithm (which will be denoted further by $ChkItem$), correspondent to the functionality func2 (II), related to automated check of answers, which will be developed in future research;
- the answer evaluation algorithm (which will be denoted further by $EvalStud$), correspondent to the functionality func3 (III), the algorithm presented in the CIM-GET model section, which represents the part of the model responsible with learning analytics and which will be described in the next section. This algorithm also uses the item prediction algorithm (which will be denoted further by $ItemPred$) and which will be presented in the last part of the section 3.

The item generation uses the algorithm GenTest in order to generate the test using methods presented before, where M is the number of items in $BD1$. A schematic approach of this algorithm is:

```
for i = 1, M do
    select items from BD1, resulting clusters Cj, j = 1, nr_kw;
    a Ti test is generated using questions from Ci;
    the Ti test is visually generated and given to the students
endfor
```

3.2. CIM-GET model

3.2.1. Model components

The CIM-GET model represents the part of DMAIR model, being one of the three main modules that were presented at the end of the previous subsection. In this matter, this model consists in the answer evaluation module, which aims the determination of the assessment performance, especially regarding the student performance for a specific item.
The CIM-GET module is designed starting from the premise that an incorrect answer to an item may indicate that the subject of the item is not fully understand, especially in certain conditions (e.g„ other items within the test are responded correctly for a student response, the item gets repeatedly wrong answers for more students etc.). In this matter, the model takes into account several factors in order to determine the direct causality between the poor understanding of the subject and the incorrect answer to an item with the respective subject.
The model needs additional variables in order to be completely described. These variables are:

- the period of time T;
- the number of tests given in the time T, M;
- the frequency of the assessment f;
- the number of the students in the group N;

The model structure consists in the existence of several components:

- the item q, described in the previous subsection, but with several additional characteristics, that will be presented in the next list;
- the test T, also described in the previous subsection, which will be enriched with several statistical indicators;
- the student result S, which contains information related to the assessment results of a specific student;
- the group of students result G, which contain statistical information related to the assessment results of a specific group of students (e.g., class, group).

Within the model, an item is considered to be correctly answered (marked with 1) or incorrectly answered (marked with 0). As a premise for the items that have fractional values of points, we will take into consideration that an item which has received equal or more than 0.5 points is marked as correctly answered and incorrectly if the value is less than 0.5 points. The additional characteristics of an item q for the CIM-GET model are:

- the scores to an item obtained by all students sc_q, stored as an array;
- the average score of the item m_q, which is the average value of all the responses of all students to the item q, taking into account also the fractional values of the scores;
- the number of correct answers l_q, which contains the number of all correct answers (marked with 1), as stated previously;
- the number of students that answered the item ta_q, which determine the number of all students that answered the item;
- the total number of attempts at_q, which stores the number of the attempts for the item q, for the case in which the teacher permits several attempts for an item;
- the average number of attempts mat_q, which stores the average number of attempts for an item q, for the case in which the teacher permits several attempts for an item;
- the standard deviation sd_q, calculated as a normal standard deviation of the item using the specific formula for standard deviation for an item q, where sc_q_i are the item scores for the item q_i, the m_q is the average score of the item q_i and N is the number of students that responded to the item:

$$sd\_q = \sqrt{\frac{\sum i = 1^N sc\_q_i - m\_q}{N}} \tag{2}$$

- the upper-lower count uc_q and lc_q, considering that the group of students is split in three groups: high (27% of students), middle (46% of students) and low (27% of students) scores; thus, uc_q equals the count of correct answers from the upper 27% of N and lc_q equals the count of correct answers from the lower 27% of N; for example, from a list of 100 answers sorted descendingly by score, uc_q would be the number of correct answers from the first 27 answers and lc_q would be the number of correct answers from the last 27 answers;
- the item discrimination d_q, $d\_q \in$ [-1, +1], which determines for an item the amount of discrimination between the responses of the upper and the lower group and which is calculated using the specific formula, where uc_q is the upper count, lc_q is the lower count and N is the number of students that responded at the item:

$$d\_q = \sqrt{\frac{uc\_q - lc\_q}{0.27 \times N}} \tag{3}$$

- the point biserial pbs_q, $pbs\_q \in$ [-1, +1], which shows whether the item is discriminating high-performing students from low-performing students, determining if the question is well written, and which is calculated as a Pearson correlation coefficient between the number of correct responses of a student to an item q and the number of all the correct answers to the other items than q in the test.

The additional characteristics for a given test T for the CIM-GET module are:

- the test length t_l, related to the number of questions in the test;
- the average score of the test m_T;

- the diversity index of the item type D_T, which shows the diversity of the item types taken into consideration (multiple-choice, short answer and essay) and which is calculated as a Simpson's Index of Diversity, as follows, where qt_m is the number of multiple-choice items, qt_s the number of short-type items and qt_e the number of essay-type items:

$$D\_T = \frac{qt\_m \times (qt\_m - 1) + qt\_s \times (qt\_s - 1) + qt\_e \times (qt\_e - 1)}{t\_l \times (t\_l - 1)} \tag{4}$$

The characteristics of a student results component S are:

- the total score of a student to all tests tt_S;
- the average score of a student to all tests mt_S;
- the total score of a student to individual tests t_S;
- the average scores of a student to individual tests m_S;
- the total score of a student to the items of the same subject ts_S.

The characteristics of a group results component G are:

- the average score of a group to all tests mt_G;
- the average scores of a group to individual tests m_G.

3.2.2. Model functionality

The model has a simple premise and is built on the generation phase of the items. Shortly, after the $GenTest$ algorithm is applied, the evaluation of the items is made, using the methodology described previously for $EvalStud$. A visual representation of the model can be seen in Figure 3.

The functionality of the model consists in the actions that can be performed within the model. The main two actions consist in:

1. the determination of the subjects that need to be revised based on the answers given by the students to the periodic assessment, denoted by $EvalStud$;
2. the determination of the probability of an item to be correctly answered by a student or by a group using k-means clustering, denoted by $ItemPred$.

The algorithm $EvalStud$ consists in the navigation of the following steps:

1. The students log in and solve the tests.
2. For each student and a specific test, a report is generated, created by following the next steps:
   a) The items that have obtained lower values of m_q and l_q are filtered.
   b) The values of the item parameters d_q, pbs_q, ta_q, dd_q, D_T, ts_S are verified.
   c) The subjects of the items are then extracted and verified to have obtained lower values for m_q and l_q in other items with the same subject for a large number of students.
3. The subjects of the items that validate the rule presented in substep 2c) are output.
4. The reports are introduced in a dataset of reports, referred further as $BD2$.

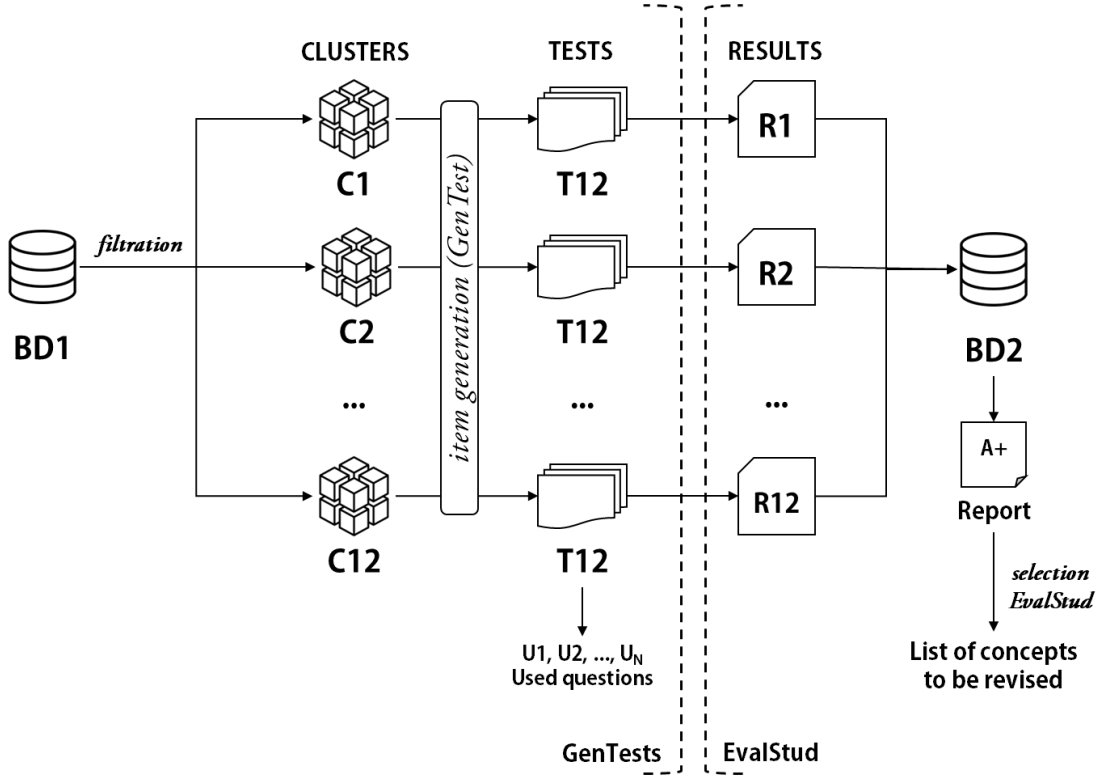

Figure 3: Visual depiction of the CIM-GET model.

A schematic approach of this algorithm is:

```
for i = 1, M do
    for j = 1, N do
        student Sj solve test Ti;
        report Ri is generated for Sj;
        Ri is introduced in BD2
    endfor
endfor
```

The algorithm $ItemPred$ consists in applying a k-means clustering to the set of data, which will be a training set for the algorithm, in order to determine whether an item is likely to be answered correctly / incorrectly for a specific student. In this matter, two clusters are formed, $Correct$ and $Incorrect$, correspondent to the probability of an item to be responded correctly / incorrectly by a student. The training data will consist in the next values: dd_q, l_T, t_q,N, mt_S, m_S. This algorithm will be extended in a further research.

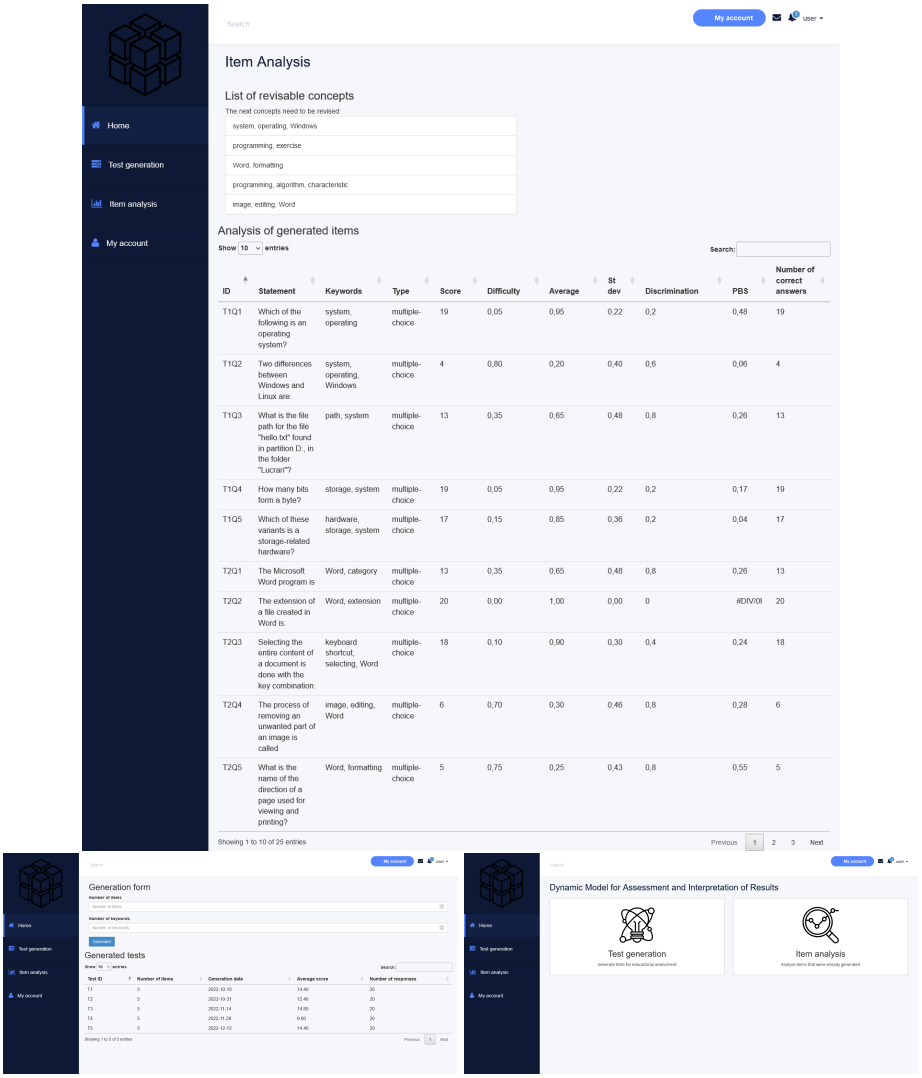

Figure 4: Screenshots of the implementation of the CIM-GET model.

4. Implementation and results

An implementation of the $GenTest$ part of the DMAIR model has been made and it is presented in previous research, such as the one described in [29].
The implementation was made using the PHP web programming language and the interface was created using the Bootstrap library, which is based on HTML, CSS and JavaScript languages. A representation of the interface for the generation of tests component and the item analysis component is shown in the next figure.

As for the results obtained related to the implementation, a specific context with several parameters was considered. The item dataset $BD1$ is not presented in this paper due to the large amount of data, but it is available in a repository [30]. The tests and items related to them

Table 1
The items taken into account and their characteristics

ID	Statement	Keywords	dd_q
T1Q1	Which of the following is an operating system?	system, operating	0,05
T1Q2	Two differences between Windows and Linux are:	system, operating, Windows	0,80
T1Q3	What is the file path for the file "hello.txt" found in partition D:, in the folder "Lucrari"?	path, system	0,35
T1Q4	How many bits form a byte?	storage, system	0,05
T1Q5	Which of these variants is a storage-related hardware?	hardware, storage, system	0,15
T2Q1	The Microsoft Word program is	Word, category	0,35
T2Q2	The extension of a file created in Word is:	Word, extension	0,00
T2Q3	Selecting the entire content of a document is done with the key combination:	keyboard shortcut, selecting, Word	0,10
T2Q4	The process of removing an unwanted part of an image is called	image, editing, Word	0,70
T2Q5	What is the name of the direction of a page used for viewing and printing?	Word, formatting	0,75
T3Q1	A program that is used to view websites is called:	browser, Internet	0,05
T3Q2	What is the term for unsolicited emails?	Internet, email	0,05
T3Q3	TCP/IP is:	protocol, Internet	0,30
T3Q4	URL means:	browser, URL, Internet	0,45
T3Q5	Which of the following variants is not a browser?	browser, Internet	0,45
T4Q1	The step-by-step procedure for solving a problem is called:	programming, algorithm	0,20
T4Q2	This characteristic of algorithms often draws the line between what is feasible...	programming, algorithm, characteristic	0,75
T4Q3	A water lily covers the surface of the water in 30 days. How many days do it...	programming, exercise	0,45
T4Q4	What is the result of the expression: (5 > 7) AND (0 < 2 * 5 < 15)?	programming, boolean, exercise	0,55
T4Q5	What is the minimum number of comparisons to sort ascendingly ...?	programming, exercise, sort	0,80
T5Q1	What is the name of the intersection of a column and a row in a worksheet?	Excel, row, line, cell	0,05
T5Q2	What function in Excel returns the sum of a number range?	function, Excel, sum	0,20
T5Q3	The process of arranging the elements of a column in a particular ...	sort, Excel	0,55
T5Q4	In Excel, the rows are numbered with:	Excel, row, line, cell	0,40
T5Q5	Which function in Excel returns the average of a range of numbers?	function, Excel, average	0,20

will be presented in Table 1.
The initial context was considered to be formed of a group of 20 students which participated to an ICT course for a period of a semester (14 weeks) and a number of 5 tests was given during this period. Each test was generated in order to contain 5 questions with specific subjects related to the usage of various applications (Word, Excel) or notions regarding Internet, programming and operating systems. The type of all questions was multiple-choice. The columns presented in Table 1 show the item unique identifier (id_q), the item statement (st_q), the item list of keywords (kw_q) and the degree of difficulty of the item (dd_q).

For the items described in Table 1, the responses were analyzed by determining the values of the parameters of the model taken into account. For this specific example, the score was equal to the correct number of responses, due to the fact that every question had a score of 1 point. The results are shown in Table 2. The columns presented in Table 1 show the degree of difficulty (dd_q), the standard deviation (sd_q), the item discrimination (d_q), the point-biserial (pbs_q), the average score m_q) and the number of correct answers (m_q).

After the responses, several items were determined as being more difficult than the others and the list of the subjects which can be revised that was obtained after the analysis of the

Table 2
The analysis results of the items

	Score	dd_q	sd_q	d_q	pbs_q	m_q	l_q
T1Q1	19	0,05	0,22	0,2	0,48	0,95	19
T1Q2	4	0,80	0,40	0,6	0,06	0,20	4
T1Q3	13	0,35	0,48	0,8	0,26	0,65	13
T1Q4	19	0,05	0,22	0,2	0,17	0,95	19
T1Q5	17	0,15	0,36	0,2	0,04	0,85	17
T2Q1	13	0,35	0,48	0,8	0,26	0,65	13
T2Q2	20	0,00	0,00	0,0	-	1,00	20
T2Q3	18	0,10	0,30	0,4	0,24	0,90	18
T2Q4	6	0,70	0,46	0,8	0,28	0,30	6
T2Q5	5	0,75	0,43	0,8	0,55	0,25	5
T3Q1	19	0,05	0,22	0,2	0,40	0,95	19
T3Q2	19	0,05	0,22	0,2	0,16	0,95	19
T3Q3	14	0,30	0,46	1,0	0,56	0,70	14
T3Q4	11	0,45	0,50	1,0	0,43	0,55	11
T3Q5	11	0,45	0,50	0,4	-0,16	0,55	11
T4Q1	16	0,20	0,40	0,4	-0,03	0,80	16
T4Q2	5	0,75	0,43	0,2	-0,22	0,25	5
T4Q3	11	0,45	0,50	0,8	0,16	0,55	11
T4Q4	9	0,55	0,50	0,8	0,22	0,45	9
T4Q5	4	0,80	0,40	0,4	0,11	0,20	4
T5Q1	19	0,05	0,22	0,2	0,37	0,95	19
T5Q2	16	0,20	0,40	0,4	0,30	0,80	16
T5Q3	9	0,55	0,50	0,8	0,18	0,45	9
T5Q4	12	0,40	0,49	0,8	0,00	0,60	12
T5Q5	16	0,20	0,40	0,6	0,30	0,80	16

results contains topics such as operating systems, Windows OS, programming, Microsoft Word, formatting, algorithm, algorithm characteristics and practical applications related to programming. In this matter, the number of items that were selected was approximately 27% of the total number of items. The selection of the items was made based on a threshold which has a statistical meaning, as in the case of the upper-lower count, which is the 27% from the total number of items, or, in case of large sets of items, the items that obtained a score lower than 27% of the maximum score of the test. The items that generated these revisable topics were Q2 from Test 1, Q4 and Q5 from Test 2 and Q2 and Q5 from Test 4, which obtained the lowest number of correct responses.

The item analysis confirmed that the mentioned items had the highest degree of difficulty. The other parameters related to the validity of the test showed that the majority of the questions were designed properly. Related to each parameter, the next results were obtained:

- The item discrimination (d_q) showed that the majority of items which had a lower degree of difficulty were not good discriminators, while the more difficult ones discriminated better between the best scores and the lower ones, which is indicated in an assessment test.

Table 3
The items that were selected and their characteristics

	Score	dd_q	sd_q	d_q	pbs_q	m_q	l_q
T1Q2	4	0,80	0,40	0,6	0,06	0,20	4
T2Q4	6	0,70	0,46	0,8	0,28	0,30	6
T2Q5	5	0,75	0,43	0,8	0,55	0,25	5
T4Q2	5	0,75	0,43	0,2	-0,22	0,25	5
T4Q5	4	0,80	0,40	0,4	0,11	0,20	4

- The point-biserial coefficient shows that several item can be improved in order to form a well-designed test. The values below 0.1 show items that can be improved as discriminators, especially for those that had higher degrees of difficulty.
- The score and the degree of difficulty and also the standard deviation were altogether correlated (the items with specific values of sd_q between 0.40 and 0.46 were the ones which had the lowest score and the highest degree of difficulty).

5. Conclusions

The most important part of the assessment performance is related to the good design of the assessment test. In this matter, the implementation of this model provides a really useful tool for a good design of the items of the test and, in the same time, provides information related to topics that can be revised during a period of time in an educational context. In this matter, this implementation can be extremely helpful for the determination of the subjects that need additional time for teaching and understanding. The model shows to be a viable one, due to the nature of the issue that responds to and the methods used to solve this issue. Given the fact that assessment is currently one of the most researched topics in education, the model and its implementation can be considered as important in order to obtain a well-designed test, after the implementation can be scaled for more general environments.
Regarding the issue of the determination of the revisable topics during an educational period of time based on the assessment, the traditional approach of item analysis was a starting point that allowed both the usage of proven scientifical tools related to the analysis of the items and the responses and the checking tool to validate the results obtained using the approach from the CIM-GET model. In this matter, statistical data resulted from the item analysis approach has proven to be a standard validator for the methods used for the described model.
As future work, the model will be improved with an automatic answer checking tool and also with the refinement of the tools presented in the paper. In the same time, the semi-automated aspect of the model will be transformed to an automated one in future research papers. Also, the implementation and documentation of the DMAIR model will be completed and described in further research, leading to the possibility of the usage of an assessment tool which can provide useful and accurate results for the assessment process.

References

[1] S. L. Craig, S. J. Smith, B. Frey, Professional development with universal design for learning: supporting teachers as learners to increase the implementation of udl, Professional Development in Education 48 (2019) 22 – 37. doi:https://doi.org/10.1080/19415257.2019.1685563.

[2] L. R. Ketterlin-Geller, Knowing what all students know: Procedures for developing universal design for assessment, The Journal of Technology, Learning and Assessment 4 (2005). URL: https://ejournals.bc.edu/index.php/jtla/article/view/1649.

[3] D. Clow, An overview of learning analytics, Teaching in Higher Education 18 (2013) 683–695. URL: https://doi.org/10.1080/13562517.2013.827653. doi:10.1080/13562517.2013.827653. arXiv:https://doi.org/10.1080/13562517.2013.827653.

[4] L. Bokander, Psychometric Assessments, Taylor 38; Francis, 2022, p. 454–465. URL: http://urn.kb.se/resolve?urn=urn:nbn:se:hj:diva-58770. doi:10.4324/9781003270546-36, [ed] S. Li, P. Hiver 38; M. Papi.

[5] T. Moses, A Review of Developments and Applications in Item Analysis, 2017, pp. 19–46. doi:10.1007/978-3-319-58689-2_2.

[6] M. Webb, D. Gibson, A. Forkosh-Baruch, Challenges for information technology supporting educational assessment, Journal of Computer Assisted Learning 29 (2013) 451–462. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/jcal.12033. doi:https://doi.org/10.1111/jcal.12033. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/jcal.12033.

[7] S. Z. Siddiqui, Framework for an effective assessment: From rocky roads to silk route, Pakistan Journal of Medical Sciences 32 (2017) 505 – 509. doi:10.12669/pjms.332.12334.

[8] A. Ben-Simon, R. Bennett, Toward more substantively meaningful automated essay scoring, Journal of Technology, Learning, and Assessment 6 (2007) 4–44.

[9] P. Deane, On the relation between automated essay scoring and modern views of the writing construct, Assessing Writing 18 (2013) 7–24. URL: https://www.sciencedirect.com/science/article/pii/S1075293512000451. doi:https://doi.org/10.1016/j.asw.2012.10.002, automated Assessment of Writing.

[10] J. Gardner, M. O'Leary, L. Yuan, Artificial intelligence in educational assessment: 'breakthrough? or buncombe and ballyhoo?', Journal of Computer Assisted Learning 37 (2021) 1207–1216. doi:https://doi.org/10.1111/jcal.12577.

[11] B. Das, M. Majumder, S. Phadikar, A. A. Sekh, Multiple-choice question generation with auto-generated distractors for computer-assisted educational assessment, Multimedia Tools and Applications 80 (2021) 1–19. doi:10.1007/s11042-021-11222-2.

[12] S. K. Saha, D. R. CH, Development of a practical system for computerized evaluation of descriptive answers of middle school level students, Interactive Learning Environments 30 (2019) 215–228. URL: https://doi.org/10.1080/10494820.2019.1651743. doi:10.1080/10494820.2019.1651743. arXiv:https://doi.org/10.1080/10494820.2019.1651743.

[13] B. Zou, P. Li, L. Pan, A. T. Aw, Automatic true/false question generation for educational purpose, in: Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022), Association for Computational Linguistics, Seattle,

Washington, 2022, pp. 61–70. URL: https://aclanthology.org/2022.bea-1.10. doi:10.18653/v1/2022.bea-1.10.
[14] B. Das, M. Majumder, Factual open cloze question generation for assessment of learner's knowledge 14 (2017). doi:10.1186/s41239-017-0060-3.
[15] A. Malafeev, Automatic generation of text-based open cloze exercises, volume 436, 2014, pp. 140–151. doi:10.1007/978-3-319-12580-0_14.
[16] H. Ali, Y. Chali, S. A. Hasan, Automatic question generation from sentences, in: Actes de la 17e conférence sur le Traitement Automatique des Langues Naturelles. Articles courts, ATALA, Montréal, Canada, 2010, pp. 213–218. URL: https://aclanthology.org/2010.jeptalnrecital-court.36.
[17] X. Zheng, Automatic question generation from freeform text, 2022. doi:10356_163315.
[18] C. Diwan, S. Srinivasa, G. Suri, S. Agarwal, P. Ram, Ai-based learning content generation and learning pathway augmentation to increase learner engagement, Computers and Education: Artificial Intelligence 4 (2023) 100110. URL: https://www.sciencedirect.com/science/article/pii/S2666920X22000650. doi:https://doi.org/10.1016/j.caeai.2022.100110.
[19] S. Burrows, I. Gurevych, B. Stein, The Eras and Trends of Automatic Short Answer Grading, Artificial Intelligence in Education 25 (2015) 60–117. URL: http://link.springer.com/article/10.1007/s40593-014-0026-8. doi:10.1007/s40593-014-0026-8.
[20] M. J. A. Aziz, F. D. Ahmad, A. A. A. Ghani, R. Mahmod, Automated marking system for short answer examination (ams-sae), 2009 IEEE Symposium on Industrial Electronics & Applications 1 (2009) 47–51.
[21] V. Zhong, W. Shi, W.-t. Yih, L. Zettlemoyer, Romqa: A benchmark for robust, multi-evidence, multi-answer question answering, 2022. URL: https://arxiv.org/abs/2210.14353. doi:10.48550/ARXIV.2210.14353.
[22] D. R. Ch, S. K. Saha, Automatic multiple choice question generation from text: A survey, IEEE Transactions on Learning Technologies 13 (2020) 14–25.
[23] G. Rasch, An individualistic approach to item analysis, Readings in mathematical social science (1966) 89–108.
[24] A. K. Hussein, A. M. A. Al-Hussein, Testing & the Impact of Item Analysis in Improving Studentsâ€™ Performance in End-of-Year Final Exams, English Linguistics Research 11 (2022) 30–36. URL: https://ideas.repec.org/a/jfr/elr111/v11y2022i2p30-36.html.
[25] M. R. Novick, The axioms and principal results of classical test theory, Journal of Mathematical Psychology 3 (1966) 1–18. doi:http://dx.doi.org/10.1016/0022-2496(66)90002-2.
[26] D. J. Weiss, M. E. Yoes, Item response theory, Advances in Educational and Psychological Testing: Theory and Applications (1991) 69–95. doi:http://dx.doi.org/10.1007/978-94-009-2195-5\textunderscore3.
[27] R. K. Hambleton, R. W. Jones, An ncme instructional module on comparison of classical test theory and item response theory and their applications to test development, Educational Measurement: Issues and Practice 12 (1993) 38–47. doi:http://dx.doi.org/10.1111/j.1745-3992.1993.tb00543.x.
[28] G. Abdelrahman, Q. Wang, B. Nunes, Knowledge tracing: A survey, ACM Comput. Surv. 55 (2023). URL: https://doi.org/10.1145/3569576. doi:10.1145/3569576.

[29] D. A. Popescu, N. Bold, The development of a web application for assessment by tests generated using genetic-based algorithms, CEUR Workshop Proceedings (2016).
[30] N. Bold, Item Dataset, https://github.com/nicolaebold/cim_get, 2023.