Tim Lutz (2021)
Automatic evaluable test of the algebra knowledge of first-year students
In: Contributions to the International Meeting of the STACK Community 2021.
TTK University of Applied Sciences: Tallinn, Estonia.
DOI: 10.5281/zenodo.4916033

Automatic evaluable test of the algebra knowledge of first-year students

Tim Lutz[1]

Abstract: Untrained students do not yet know how to enter solutions in STACK. In this article, an automated pre-processing is presented. With the help of the pre-processing, even inexperienced students can perform an automated diagnostic test of elementary algebra with high recognition rates in STACK. Why pre-processing can be an important tool in automated evaluation is demonstrated by the example of a task of "understanding the letters" according to Küchemann. For this purpose, data collected in the aldiff project of the Heidelberg University of Education is analyzed; aldiff develops an automated test of elementary algebra knowledge.

Keywords: automated test evaluation, algebra, answer pre-processing, test, first-year students

1 Theory and background

1.1 Küchemann's "letter as a specific unknown"

Küchemann describes 6 categories of students' understandings of the letters:

"letter evaluated", "letter not used", "letter used as an object": For the most part, these categories do not occur among first-year students in the aldiff project.

"letter used as a specific unknown", "letter used as a generalised number", "letter used as a variable": These three categories are slightly more interesting for the aldiff project, as they are more likely to apply to the target group of first-year students. For this paper, the category "used as a specific unknown" is of particular importance: "Children regard a letter as a specific unknown number, and can operate upon it directly." [Kü81]

Applied to a task, it means that a student relates letters in the task to numbers: The letter stands for a clearly defined number, which is still unknown to me. Knowing that this is a number, I can already operate with the letter [Lu21a].

[1] Universität Landau, Institut für Mathematik, Fortstraße 7, Landau, 76829, tim.lutz@alumni.uni-heidelberg.de

This sets in motion the following cascade of understanding:

1. The user has recognized: There is a letter. The letter stands for a very specific number.

2. Therefore, the student now feels the urge to find out that number: "I try to find out the number because I assume that it is a specific one. I can determine it; so I should determine it."

1.2 The task "What can you say about r if r = s+t and r+s+t = 30"

The task "What can you say about r..." is categorized by Küchemann as a diagnostic task for "used as specific unknown". Küchemann deliberately makes the question vague. It is not specified that a correct answer can start with "r =…". Nor is it said that the answer can be given in the form of an equation. Küchemann does not ask to "calculate" or "determine" r. With this hint he would already give a part of the interpretation.

In the open question format "What can you say about …" the request to determine r is hidden. But because the request is not explicitly expressed, it is not given how one should answer the question. To the person who can recognize the letter as "used as a specific unknown", the target "r = 15" is quickly discovered. How exactly the respondent gives the answer, whether simply as "15" or as a text-based response, must be left open in this deliberately vague question format.

This task is also used in the aldiff study in its German version according to Oldenburg [Ol09]. The task is answered in many different ways by students in project aldiff. Some examples are shown in Table 1. For easy reference in the text see the id in the left column. The answers given in German language are translated in column "Response". The column "indicator of "used as a specific unknown"" notes whether the answer was assigned to this category in the aldiff project.

id	Response	Indicator of "used as a specific unknown"
(a)	15	X
(b)	r=15	X
(c)	r is 15.	X
(d)	r must be 15	X
(e)	r=30	(X)
(f)	r=s+t	--
(g)	30 = 2*s + 2*t	--
(h)	r is twice t (or s)	(X)

Tab. 1: example responses. X: is indicator for "used as a specific unknown".

(a), (b), (c), (d): for these responses of type "r = 15", it is assumed in aldiff that a variable understanding of "used as a specific unknown" has been reached by the student.

The subject is able to interpret the vague query in a mathematical way and proceeds down the path to the solution until “r” is specifically determined. Thinking further, if “r” were not determinable in another example, it is to be expected that a respondent who is only at this level would react confused that a final calculation was not possible.

(e): Although the answer “r = 30” is wrong, one could also recognize a failed attempt to determine “r” in this. And thus, an assignment to “used as a specific unknown” cannot be ruled out in principle. In this case, it is assumed in aldiff that the subject is not confident enough in processing tasks that involve letters “used as a specific unknown”.

(f): Although the answer “r = s + t” resembles the answer (b) at first sight, it is only a repetition of the statement given in the task. The vague suggestion of the task does not push the student to search for a more specific solution. Consequently, this answer is evaluated as an indicator in the project aldiff for not-“used as a specific unknown”.

(g): Basically, all answers that do not indicate the specific determination of “r” are evaluated as false and as indicators for not-“used as a specific unknown”. In (g), “r” is not mentioned. No matter what the intent is behind answers like (g), the student is obviously not focused on determining “r” in his response.

(h): “r” remains undetermined. The subject probably has the wrong thought process: “r is as big as the two letters together.” “So r is twice as large as the two.”

(h) as interpretation of “wrong thinking for advanced”: “r” remains undetermined. Here the student probably thought in a wrong way: “If r = s + t, then r is larger than t” and “if s and t were equal, then r would be twice as large as t.” “So, r is (every time) twice t.” In this case of interpretation, it can be assumed that this is an erroneous further development of the category “letter evaluated”. In the category “letter evaluated”, the student originally sets r equal to a number e.g. “r=3” to see what happens. In the further development of this approach, additional assumptions made up by the user are now added, such as “s could be equal to t. Let's see what happens next in the task.” More empirical research is required to determine to what extent test persons proceed in such a way that they mix up levels of variable understanding.

It also suggests: levels of understanding of the letters can be reached unevenly. Either the student has failed to determine “r” and writes down what he has found out so far, or more likely he is satisfied with the answer because he guesses (but indeed did not reach) the level of the letter “used as a generalised number”, ignoring that here the variable can and therefore should be understood as “used as a specific unknown”.

It does not matter whether correct or incorrect statements are made in this and similar ways: The respondent is satisfied with the answer or does not manage to convert the statement into specific numbers. This behavior is also evaluated as an indicator for insufficient achievement of the "specific unknown" level.

2 Theoretical framework aldiff; an automated test of algebra

The test items of aldiff were collected and developed in the predecessor project based on an extensive literature review [PDV13], [LPV18].

Emphasis was also placed on a very mixed task format. The test items e.g. include multiple choice and timed items; however, many of the items are asked as open text items and mostly require the naming of a specific algebraic expression, e.g. "n=4". The responses to these tasks were collected digitally in the aldiff study [Lu21b].

2.1 Necessity of input instructions

The use of STACK [Sa13] requires input instructions. Input instructions are required for several reasons.

- The input of mathematical characters: e.g. the student knows "2^3 is written as 2^3".
- The input of specific mathematical objects: e.g. the subject knows that "an equation is expected by the system as an input object".

In aldiff, all tasks were initially asked as questions without automatic scoring. The students entered their responses mostly in open text fields, following general input instructions of entering mathematical characters.

In a few STACK tasks, the input of certain mathematical objects was often treated by the students in the style of dealing with non-digital tasks. This behavior is attributed to the fact that it can be assumed that the students had little or no previous experience with the digital input of mathematical responses. Especially those tasks are affected which, due to the task format, additionally encourage such answers with parts of natural language, like the task by Küchemann presented here.

In aldiff, a short compressed version of the test was created based on the data from the study. This can be evaluated automatically with STACK. In the testing of the STACK tasks, it was shown that even in situations in which subjects could be aware of the automatic evaluation, they nevertheless continue to answer the tasks frequently in natural language. As an example for such a typical answer type: "it is n=4".

In order to still be able to reliably evaluate with STACK in an automated way, one could now react in different ways.

2.2 Force input of a mathematical object by adjusting the input fields

Subjects are asked to enter specific mathematical objects. For example, a hint could be given in front of the input field: "The answer is an equation" or "n=".

It cannot be assumed that more general input instructions at a task-based level, such as "enter only mathematical objects" or "enter only an equation" would be processed correctly by the subjects in principle.

2.3 Pre-processing, which tries to convert inputs into a format that can be read by STACK

The task is not changed. The input is processed by a script and then passed on to STACK.

It makes sense to choose the first option (2.2) in situations where subjects are to learn to work with STACK and in more complex tasks, where the difference in the answer format cannot be considered to have any bearing on the competence of the task processing.

However, in the diagnostic test developed in aldiff, the students should not be required to learn *how to use* STACK and the competence to solve the task *is related to the input object* in a didactically relevant way, which is made clear by the analysis of the example inputs above.

Answers like (g) “30=2*s+2*t” should still be possible as an input. The subject has operated with the two equations. This interaction does not happen purposefully enough. The student should notice that he does not make a statement about “r”, although he was just asked to do so. Moreover, STACK can even process this response, so that responses of this kind should not be prevented in the interest of diagnostics, for example by placing “r=” in front of the input field.

In contrast, solutions of type (h) “r is twice t” in the presented task do not necessarily have to be evaluated by STACK. By default, they are evaluated as incorrect by the system because these answers cannot be processed in a system like STACK [Lu21c].

However, text responses of this type never indicate correct processing in the sense of “r=15” in aldiff. Thus, in order to make the main distinction “used as a specific unknown” vs. not-“used as a specific unknown” in this task, STACK does not need to be able to process long text responses like (h) at all.

Pre-processing has to reduce false negative examples (a),(b),(c),(d). Additionally feedback to faulty algebraic expressions can be done in feedback trees, such as (e),(f), and (g). Outputs that are not evaluable by STACK despite pre-processing are then most likely correctly classified as not “used as a specific unknown”.

There are tasks in the aldiff study where usually neither true positives nor true negatives can be converted to a STACK readable format with pre-processing. In such cases, it is not possible to argue as described above: the diagnostic function of the task would be lost.

It is shown for one of these empirically studied tasks how machine learning can take over the functionality of categorizing the responses in predefined categories [Lu21a].

Summed up: even small changes in the task definition with regards to the input format in elementary tasks, such as the default value “n=”, can lead to undesirable effects. Consequently, responses like: (a),(b),(c),(d) must be recognized by STACK to reduce false negatives to a minimum without creating false positives by pre-processing.

How can task responses such as (a),(b),(c),(d), which in part typically contain small portions of natural language, be processed with STACK?

An analysis of the task responses with N=407 should show which frequent elements of natural language occur. With the goal of not changing the content, but only making it readable for STACK through substitution and omission, suggestions for the script will then be developed.

The resulting pre-processing script is then evaluated against test cases of N=60 responses not yet considered in the pre-processing considerations to provide an assessment of the effectiveness of pre-processing.

3 Results

The analyses show that even very few rules in pre-processing are sufficient to allow automatic evaluation and reliable differentiation between indicators for “used as a specific unknown” and not-“used as a specific unknown”.

While pre-processing is similar for all expected answers of the type "equation", it must be adapted for the different tasks. To show that compatibility with STACK can also be established for similar tasks by analogous pre-processing, the example “n=4” is chosen.

For the expected answer of the type “equation”, task-specific typical answers are entered by subjects:

If n is 4. // It is 4 // It is n=4

For pre-processing it is, in principle, sufficient to delete certain words and to replace other words depending on the situation.

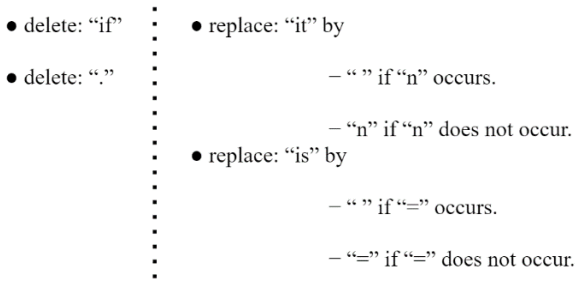

Fig. 1: pre-processing of expected answer type “n=4”

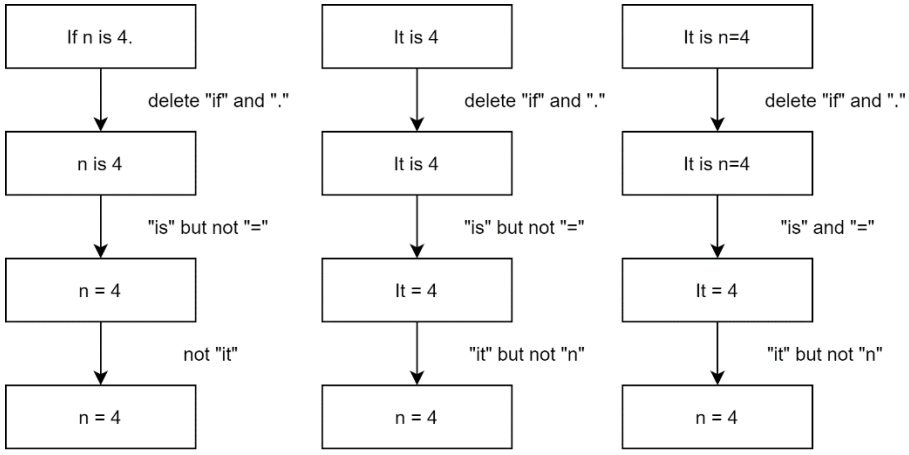

Fig. 2: pre-processing examples

With the application of these simple rules, a task that was originally only 85% evaluable (automated coding matches manual coding) could now reach up to 100% in the validation dataset. Similar sets of pre-processing rules were developed for point coordinate notations. All rules are customized to the tasks according to the aldiff data.

By default, the answer is processed invisibly for the respondent before being passed on to STACK in order to influence the respondent as little as possible. Alternatively, for possible uses outside of a diagnostic test, the respondent can be informed before submission that a change in input will be made and it is explained to him why only mathematical objects should be entered in STACK fields. This is helpful when getting subjects used to the STACK task format.

4 Outlook

For the tasks in aldiff that contain small portions of natural language, the procedure described in this article is sufficient to raise the automatic classification of responses for the diagnostic test to at least 95% (in the validation data).

However, individual tasks in aldiff are fundamentally unsuitable for evaluation with STACK. In tasks where short explanations and justifications are required, other approaches to automated evaluation must be chosen. In such cases, machine learning models can help to perform an automated evaluation nevertheless [Lu21a] [Lu21d].

The algebra test as a short compressed version of the test developed in aldiff will soon be made available as a STACK version via an e-learning platform.

Information about aldiff (and much more): https://tim-lutz.de/test/algebra

Bibliography

[Kü81] Küchemann, D.: Chapter 8: Algebra. In (Hart, K.M. ed.): Children's understanding of mathematics: 11-16, John Murray, London, pp. 102-119, 1981.

[Lu21a] Lutz, T.: Machine Learning Model for Automated Text Classification of Mathematical Tasks, 2021.

[Lu21b] Lutz, T.: Diagnose und Förderung in der elementaren Algebra. Entwicklung eines Diagnoseinstrumentes und Vorbereitung eines Förderkonzeptes, Springer, 2021.

[Lu21c] Lutz, T.: Automatisiertes Feedback in Echtzeit für die Arbeit mit physischen Materialien und ikonischen Darstellungen unter Verwendung von Machine Learning und AR. In Beiträge zum Mathematikunterricht, 2021.

[Lu21d] Lutz, T.: Stellungnahme: Algorithmus und seine Bedeutung für die digitale Mündigkeit. Algorithmen bestimmen unsere Welt. Lass dich nicht von Algorithmen bestimmen. Über die Magie von Algorithmen und ihre Entmystifizierung. https://tim-lutz.de/algorithmen-entmystifizieren/

[LPV18] Lutz, T.; Pinkernell, G.; Vogel, M.: Diagnose und Förderung im Bereich der elementaren Algebra an der Schnittstelle Übergang Schule-Hochschule. In Beiträge zum Mathematikunterricht, 2018.

[Ol09] Oldenburg, R.: Structure of Algebraic Competencies. In (Durand-Guerrier, V.; Lavergne, S.; Arzarello, F. ed.): Proceedings of CERME 6, Institut National de Recherche Pedagogique., Lyon, pp. 579-588, 2009.

[OHK13] Oldenburg, R.; Hodgen, J.; Küchemann, D.: Syntactic and Semantic Items in Algebra Tests. A Conceptual and Empirical View. 2013.

[PDV13] Pinkernell, G.; Düsi, C.; Vogel, M.: Aspects of Proficiency in Elementary Algebra. In Proceedings of CERME 10, Dublin, 2018.

[Sa13] Sangwin, C.J.: Computer aided Assessment of Mathematics, Oxford Univ. Press, Oxford, 2013.