

Question Answering in STACK Applying String Similarity

<https://doi.org/10.3991/ijet.v17i23.35893>

Achim Eichhorn¹(✉), Andreas Helfrich-Schkarbanenko²

¹Orientation and Undergraduate Studies Central Academic Institution, Esslingen
University of Applied Sciences, Esslingen am Neckar, Germany

²Faculty of Mechanical Engineering and Mechatronics, Karlsruhe University
of Applied Sciences, Karlsruhe, Germany
Achim.Eichhorn@hs-esslingen.de

Abstract—We present a method to evaluate fill-in-the-blank student answers in STACK using a string metric, which is not possible in the current version of STACK. To increase the quality of the evaluation, we use whitelist and blacklist instead of a single teacher answer. The performance of a STACK question equipped with a string metric is quantitatively demonstrated by evaluating its use in mathematics courses.

Keywords—online assessment, question answering, STACK, CAS, Damerau-Levenshtein edit distance, string metric, string similarity, e-learning, natural language processing

1 Introduction

Online assessment is a quick, practical and cheap way to capture the current level of knowledge of learners and to support learning processes. With the existing technical tools, however, the open or fill-in-the-blank questions can hardly be implemented, since typing errors and synonyms have to be taken into account when evaluating the student answers, see Section 2. However, this type of questions is considered important from a didactic point of view. For online assessment in the mathematics lectures at Esslingen University of Applied Sciences, we use the STACK plug-in within the Moodle learning platform, that is the world-leading open-source online assessment system for mathematics and STEM fields [8]. It is available for Moodle, ILIAS and as an integration through the web services protocol Learning Tools Interoperability. Currently there is no evaluation of the string inputs in STACK. Here we present how we have extended STACK with this missing functionality, Section 3. To achieve this we used one of the string metrics for measuring the distance between two strings: the Damerau-Levenshtein distance, which plays an important role in natural language processing. Informally, this distance is the minimum number of single-character edits (insertion, deletion, substitution, transition) required to change one string sequence into the other. This enables a string evaluation. String based distance is much easier to determine than semantics

based distance, for which one uses e.g. WordNet [2], i.e. a large database of words linked by semantic and lexical relations, equipped with a semantic metric to introduce semantic similarity. Machine learning methods, see e.g. [9], would also be applicable here, but their realization is too complex for our purposes. We implemented a string metric in computer algebra system MAXIMA, see [4], and placed the corresponding function in the *Question variables* field of the STACK question concerned. Note that the term ‘STACK question’ in this paper refers to a task. Finally, we demonstrate the performance of the proposed method by using it in a mathematics course, Section 4.

2 String inputs and the challenge of question answering

The plug-in STACK is an open source system for computer-based assessments in mathematics and related disciplines [3]. It enables task based teaching. For knowledge retrieval STACK offers a set of different input types: Algebraic Input, Checkbox, Top down list, Equivalence reasoning, Matrix, Notes, Numerical, Radio, Single character, String, Text area, True/False and Units. The inputs are evaluated – if possible – by means of MAXIMA. However, the input type ‘String’ cannot be evaluated in STACK, although it may be suitable for both closed and open questions. Closed questions are characterized by a predefined solution path and a unique solution. Questions of this type focus on factual and procedural knowledge that are relatively easy to query. In contrast, open questions are characterized by few instructions, possibly several solutions or no clear solution. According to Bloom’s taxonomy of learning objectives in the cognitive learning area, open questions are to be classified under objectives such as analyze, evaluate, create, which promote higher order thinking skills, because these questions give space for own ideas and ways of solving problems; encourage independent thinking; transfer responsibility to learners and contribute to the networking of knowledge [5]. Thus, it is especially important to be able to set open questions in an online assessment tool and to evaluate the answers. As an example, we give some questions with corresponding teacher answers, for which the input type ‘String’ is to be preferred in our opinion.

Exercise: Consider the differential equation $y'(t) + y(t) = 0$.

- a) Classify this equation.
Teacher answer: linear with constant coefficients, 1-st order, homogeneous
- b) What methods do you know that can be used to solve this DGL?
Teacher answer: exponential ansatz, separation of variables
- c) The equation above is now provided with an inhomogeneity: $y'(t) + y(t) = 1$. Which method would you apply to find the particular solution?
Teacher answer: variation of the constant
- d) Can the inhomogeneous equation be solved without the exponential approach? If yes, with which method?
Teacher answer: separation of variables

Such a task is currently not evaluable in STACK as an open question with ‘String’ as input type. However, the procedure presented in the next section sets up a possibility for this.

3 String input meets string metric

We implemented the Damerau-Levenshtein distance calculation between the *student answer* (SA) and the *teacher answer* (TA) in MAXIMA, [1, 6, 7]. To increase the quality of the assessment, we extended the function by the following components:

- We compare the student answer not only with a single teacher answer, but with a list of variations or synonyms of it. In the information technology, such a list is called *whitelist* = $\{TA_1, TA_2, TA_3, \dots\}$. Therein, for example, the frequently occurring typing errors or permutations of words can also be taken into account. Thus we obtain the distances $d(SA, TA_k)$ from which the lowest distance is used for further calculations.
- The experiments with whitelists have shown that some contrary student answers have almost equal edit distance to the teacher answer, e.g., the answers ‘asymptotically stable’ and ‘asymptotically unstable’. By introducing the so-called blacklist = $\{BA_1, BA_2, BA_3, \dots\}$. (i.e., list of incorrect answers) and determining $d(SA, BA_k)$, we succeeded in analyzing the student answers multivariately and thus being able to distinguish them more precisely, see Figure 3.
- To have a relative measure of the difference between two strings, we convert the edit distance $d(a, b) \in [0, \max\{|a|, |b|\}]$ of type integer to similarity $s(a, b) \in [0, 1]$ of type float:

$$s(a, b) := 1 - \frac{d(a, b)}{\max\{|a|, |b|\}} \quad (1)$$

where $|a|, |b|$ are the respective string lengths. For understanding we give the edit distance and similarity for the terms ‘Rectangle’, ‘Triangle’ and ‘Circle’, see Figure 1.

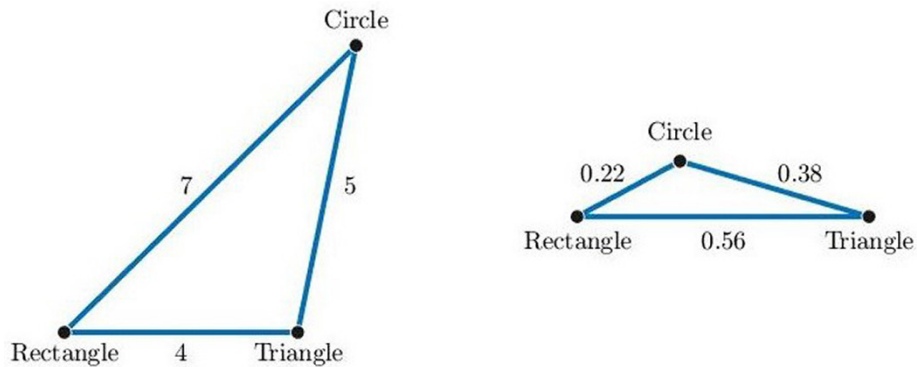


Fig. 1. Damerau-Levenshtein distance (left) and corresponding similarity (right) for the terms ‘Rectangle’, ‘Triangle’ and ‘Circle’

The high similarity of the terms ‘Rectangle’ and ‘Triangle’ is due to the common word ‘angle’. Note that for similarity (1) the triangle inequality does not hold in general.

- For the similarity between a string a and a list $list = \{b_1, b_2, \dots, b_k\}$ we use the maximum norm:

$$s(a, list) := \max_{k=1, \dots, K} (s(a, b_k)).$$

- The decision whether a student answer is ‘correct’ or ‘wrong’ is made from the similarities and the previously empirically determined threshold θ . If

$$(s(SA, whitelist) > s(SA, blacklist)) \wedge (s(SA, whitelist) > \theta), \quad (2)$$

then student answer SA is accepted, otherwise rejected, see acceptance and rejection domains in Figure 3.

The described procedure is implemented as follows.

Consider for $t > 0$ the linear, inhomogeneous differential equation

$$7 \cdot t \cdot \left(\frac{d}{dt} y(t)\right) - 21 \cdot y(t) = 6 \cdot t^8.$$

a) Which method should be used here to calculate the particular solution?

Method:

b) Give the general solution. Use the notation C for the constant.

$y(t) =$

Fig. 2. Fill-in-the-blank question a) with string as an input type

We aim to query for a suitable solution method by fill-in-the-blank question when given a differential equation, see subtask a) in Figure 2.

1. When editing a task, the edit distance function implemented in MAXIMA must be inserted in the *Question variables* field. In addition, the whitelist and blacklist are declared in it.

```
y_p: b/((n-k)*a)*t^n;
y_a: y_h+y_p;
whitelist: ["Variation der Konstanten", "Variation von Konstanten", "konstanten variieren", "Variieren Konstanten", "durch Variieren der Konstanten", "Konstantenvariation", "durch Konstantenvariation", "mittels Variation der Konstanten"];
```

2. In the *Question text* field, input variables and validation variables for string input are declared.

Consider for $(t > 0)$ the linear, inhomogeneous differential equation

$\{\{ @ eq1 @ \}$.

a) Which method should be used here to calculate the particular solution?

$\{\{ \quad \}$ Method: $[[input:ans1]]$ $[[validation:ans1]]$

3. Select ‘String’ as input type and e.g. whitelist [1] as model answer.



▼ Input: ans1

Input type		String
Model answer		whitelist[1]

4. When editing the potential response tree, the self-written MAXIMA-function `compare Strings(ans1, whitelist, true)` has to be inserted in the *feedback variables* field. This function determines the similarities $s(SA, \textit{whitelist})$ and $s(SA, \textit{blacklist})$. Then, according to condition (2), it is checked if a student answer is in the acceptance domain, see variable *textok* in the Figure. For this STACK question we set $\theta = 0.8$. The third argument of the function `compare Strings` is used for optional case-sensitivity in the calculations.

```
theta:0.8;
whiteval:compareStrings(ans1,whitelist,true)[1];
blackval:compareStrings(ans1,blacklist,true)[1];
textok:if( (whiteval>=theta) and (whiteval>blackval)) then true else false;
```

5. The boolean variable *textok* is finally used in the corresponding node of the response tree for scoring the student answer. As answer test type we choose ‘AlgEquiv’, see Figure below.

Node 1		Answer test	AlgEquiv		SAns	textok	TAns	true
--------	---	-------------	----------	---	------	--------	------	------

Herewith a STACK question is adapted and can be used in a test. The corresponding XML file is available for interested readers at this URL:

www2.hs-esslingen.de/~aeich/digitalerrueckenwind/stack-conference/downloads/String_Similarity.xml.

4 String metric in action

The STACK question shown in Figure 2 was used in the winter semester 2021/22 as part of a mini-test for the lecture Mathematics 2. It was completed by 53 students and all student answers were scored error-free. As an edit distance, we initially used the Jaro-Winkler distance \tilde{d} [11], which, however is not a metric in the mathematical sense, since it does not satisfy the triangle inequality $\tilde{d}(a, c) \leq \tilde{d}(a, b) + \tilde{d}(b, c)$. This can be verified e.g. with the triple string $a = \textit{‘Rectangle’}$, $b = \textit{‘Triangle’}$, $c = \textit{‘Angle’}$. Therefore, in the meantime, we also implemented the Damerau-Levenshtein distance, which is indeed a metric and thus more suitable for our purposes. Using student answers for

the fill-in-the-blank question in Figure 2, we demonstrate in the following the power of multivariate analysis applying the Damerau-Levenshtein distance.

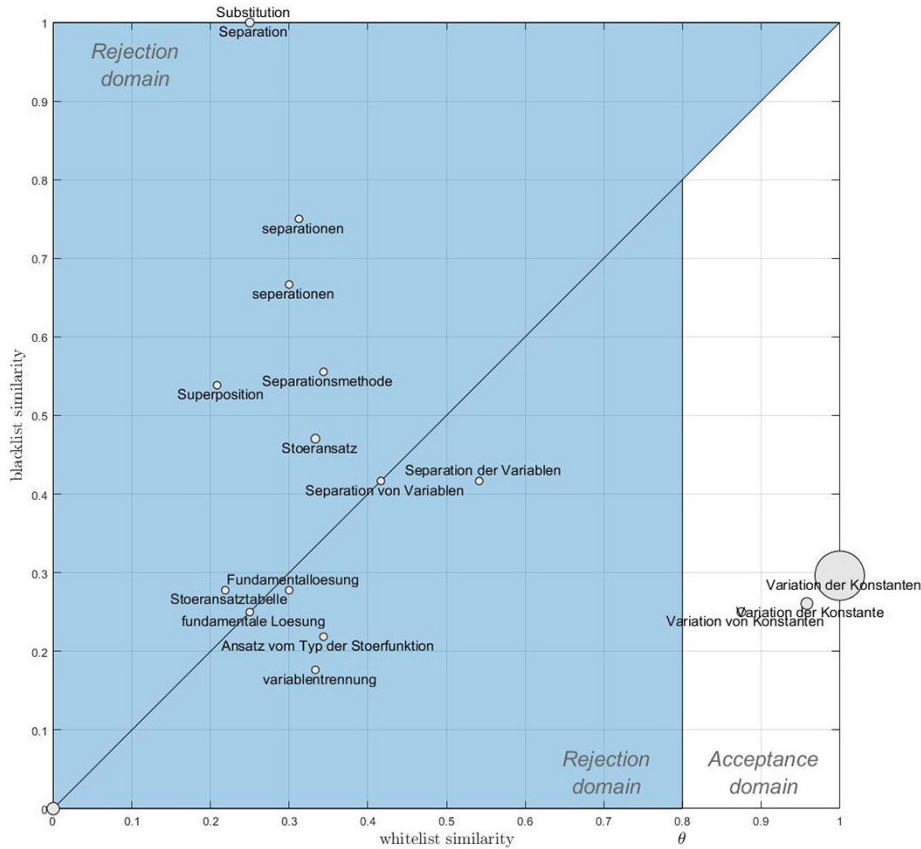


Fig. 3. Multivariate evaluation of 18 different student answers using whitelist and blacklist similarity; Acceptance domain (white) is described in (2)

In Figure 3 we see 18 different student answers (in German) which are positioned in a coordinate system according to both similarities and are classified without errors. The radii of the disks represent the number of equal student answers. In total, this task was processed 263 times. With the Damerau-Levenshtein distance, the correct answers can be isolated even with a lower threshold (compared to Jaro-Winkler distance). If $\theta = 0.8$, the condition (2) gives the white-marked acceptance domain for correct answers. For the scoring we used the *whitelist* = {‘Variation der Konstanten’, ‘Konstanten variieren’, ‘mittels Variation der Konstanten’, ‘Variieren der Konstanten’} and the *blacklist* = {‘Trennung der Veränderlichen’, ‘Substitution’, ‘Separation’, ‘Exponentialansatz’}.

4.1 Remarks

- Note that for this STACK question and the given whitelist resp. blacklist, only the consideration of the whitelist similarity would be sufficient for the evaluation. However, as mentioned above, there are situations where the blacklist is necessary.
- The Damerau-Levenshtein distance can also be computed between two longer strings, but the cost to compute it, which is roughly proportional to the product of the two string lengths, makes this impractical.
- The presented method is – strictly speaking – not only based on strings, but also on semantics, because by introducing white list and black list respectively, a (trivial) semantic graph consisting of two clusters is set up. For the sake of simplicity, the evaluation of a student answer does not take place by means of a semantic distance, but using the string distance to the respective cluster as a whole (single linkage, minimum distance, nearest neighbor, see for example [10]).

5 Conclusion

In this paper, we presented a method to evaluate the didactically important fill-in-the-blank questions in a STACK questions. For the evaluation we used e.g. the Damerau-Levenshtein distance, which calculates a distance between two strings. The determination of the distance is based on string syntax alone. To increase the reliability of the assessment, we consider both a whitelist and a blacklist of answers. The first application of the method in assessment tests was completely satisfying.

6 References

- [1] L. Boytsov, “Indexing methods for approximate dictionary searching: Comparative analysis”, *ACM Journal of Experimental Algorithmics (JEA)*, 16 (1), 2011. Available: <https://dl.acm.org/doi/abs/10.1145/1963190.1963191>. Accessed July 21, 2022. <https://doi.org/10.1145/1963190.1963191>
- [2] K. Jayakodi, M. Bandara, I. Perera, and D. Meedeniya, “WordNet and cosine similarity based classifier of exam questions using bloom’s taxonomy”, *International Journal of Emerging Technologies in Learning – 11* (4), 2016. Available: <http://www.i-jet.org>. Accessed February 21, 2022. <https://doi.org/10.3991/ijet.v11i04.5654>
- [3] The University of Edinburgh, STACK. Available: <https://www.ed.ac.uk/math/stack>. Accessed February 21, 2022.
- [4] Computer Algebra System MAXIMA. Available: <https://maxima.sourceforge.io/>. Accessed February 21, 2022.
- [5] B. S. Bloom, Editor, “Taxonomie von Lernzielen im kognitiven Bereich”, 5. Auflage. Beltz Verlag, Weinheim 1976.
- [6] F. J. Damerau, “A technique for computer detection and correction of spelling errors”, *Communications of the ACM*, 7 (3): 171–176, 1964. <https://doi.org/10.1145/363958.363994>
- [7] V. I. Levenshtein, “Binary codes capable of correcting deletions, insertions, and reversals”, *Soviet Physics Doklady*, 10 (8): 707–710, 1966.

- [8] C. Sangwin, *Computer Aided Assessment of Mathematics Using STACK* Oxford University Press, 2013. <https://doi.org/10.1093/acprof:oso/9780199660353.001.0001>
- [9] V. Sonakshi, T. Devendra, J. Amita, “A machine learning approach for automated evaluation of short answers using text Similarity based on WordNet graphs”, *Wireless Personal Communications*, 111:1271–1282, 2020. <https://doi.org/10.1007/s11277-019-06913-x>
- [10] M. von der Hude, *Predictive Analytics und Data Mining*, Springer Vieweg, 2022.
- [11] W. E. Winkler, “String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage”, *Proceedings of the Section on Survey Research Methods*. American Statistical Association, 354–359, 1990.

7 Authors

Achim Eichhorn is employed as a graduate computer scientist at the Esslingen University of Applied Sciences, Germany. He is a very experienced software developer, specialized in web technologies and interactive, multi-media e-learning content.

Andreas Helfrich-Scharbanenko is a full Professor at University of Applied Sciences in Karlsruhe, Germany. His research interests include educational technologies for learning. Co-author of a book about automatic solving of mathematical problems and generation of related teaching materials.

Article submitted 2022-09-08. Resubmitted 2022-10-23. Final acceptance 2022-10-24. Final version published as submitted by the authors.