Reliability assessment of the automatic plagiarism detection system for various editing patterns in documents containing complex mathematical notation

Agnieszka Bier, Zdzisław Sroczyński*
Faculty of Applied Mathematics, Silesian University of Technology, Kaszubska 23, 44-100 Gliwice, Poland
E-mail: agnieszka.bier@polsl.pl, zdzislaw.sroczynski@polsl.pl

Abstract. The paper presents the idea and implementation details of a novel tool intended for detection of plagiarism in mathematical content of \LaTeX files. The proposed algorithm compares \LaTeX files and detects possible similarities in mathematical formulas, providing an objective similarity score. It offers two modes of comparison: direct (symbolic) mode and verbalized math mode, where the mathematical expressions are first verbalized to spoken language form and compared as such. The solution has been tested against identified various types of plagiarism specific to mathematical symbolic notation and \LaTeX features, and proved to be an efficient tool for plagiarism detection.

1. Introduction

1.1. The problem of plagiarism
The plagiarism is the key issue for the science, technology and teaching. Although the extensive reviewer’s knowledge in the given field can possibly prevent unauthorised usage of a third party work and results, the contemporary modern world produces enormous amount of data in many publications and many media.

Therefore it is almost impossible to learn about all these achievements in a reasonable time, even for a very experienced expert. This way the thread of plagiarism increases rapidly, especially in cases which require to verify many documents in relatively short time or when the topic is wide and multidisciplinary.

The automatic algorithms detecting repetitions and suspicious similarities between documents seem to be the most appropriate solution for the issues described above. Currently, such algorithms are well designed and tested for common literature texts. They help to prevent plagiarism in the fields of history and other social sciences, but taking into account the majority of technical, mathematical and graphical documents - the challenge remains.

1.2. Plagiarism in mathematics
Mathematical textbooks, research papers and theses often contain numerous equations, symbols and mathematical formulas, which usually constitute the real value of the mathematical contents.
document. This stays in contrast to documents attributed to other disciplines, where the content is mainly given literally and may be checked for the plagiarism using algorithms developed for plain text documents.

The main problem in plagiarism detection for mathematical expressions is the fact that mathematical formulas and symbols usually require a specialized presentation form or even a separate language, which determine rules and syntax for mathematical content and may be realized in various ways. This issue becomes of particular importance, while designing an anti-plagiarism system for mathematics. One of the most popular editing systems for mathematical content is LaTeX, where all mathematical formulas and equations are encapsulated in dedicated, recognizable environments and defined by special commands corresponding to mathematical symbols and operands.

The design of a plagiarism detection tool for mathematics should also take into account possible types of plagiarism of mathematical content. This issue is tightly related with the language used for mathematical elements. In our study we have analyzed various types of plagiarism for documents written in LaTeX. As a result, we distinguished the following types:

- splitting formula into few shorter or merging few formulas into single expression or equation environment,
- reordering expressions or symbols in a formula,
- spacing adjustment (white spaces),
- equation environment changes,
- transformation of formulas by application of mathematical identities,
- transcribing parts of formulas into text equivalent form,
- replacement of symbols by other symbols (e.g. y instead of x, α instead of a, etc.).

The above plagiarism schemes were identified by the referees of Master’s and Bachelor’s degree theses as the most popular types of plagiarism regarding the mathematical content. At the same time, they are often not properly detected by the general purpose anti-plagiarism tools. Since the theses and publications at mathematical departments of universities are usually written in LaTeX, there is a need for a tool for plagiarism detection in .tex files which particularly proves its effectivity when tested against the identified types of plagiarism in math.

1.3. Existing algorithms of plagiarism detection

The papers [1, 2, 3, 4] give a comprehensive survey of conventional methods of plagiarism detection. The availability of the reference corpus used by algorithms seeking for plagiarized content is the essential complementary task, very often dependant on the national language and dialects [5, 6]. Authors of [7, 8, 9] discuss the details of various text similarity measures in big data analysis. The results of implementation of dynamic programming (Smith-Waterman algorithm) for plagiarism detection is described in [10].

Recently, there are attempts to provide the contextual searching interface for mathematical documents, which certainly provide the essential technology when it comes to plagiarism detection [11, 12]. A comprehensive overview of the methods for computing the similarity of mathematical content is given in [13]. The hybrid approach taking into account significant paraphrases and many parts of the documents, including mathematical equations compared according to histograms of symbols’ frequency is described in [14]. The basic method of the mathematical content retrieval and comparison is described in [15].

Details of the mathematical notation plagiarism versions are given in [16] with the conclusion, that equations would possibly be the core feature in the recognition of the plagiarized documents, regardless the pure text content. Therefore the basic approach of plagiarism detection is to compare pairwise equations between the tested documents and evaluate the overall similarity.
based on the level of similarity of particular pairs. This implies that the computational complexity of these kind of algorithms is of order $O(n \cdot m)$, where the number of equations in the tested documents is $n$ and $m$, respectively. Most existing solutions use the above general model for plagiarism detection, however they differ in how the particular pairs are matched and compared. Hence the overall complexity of such solution is also dependent on the choice of the equation similarity detection method used. The fine-tuning of the mathematical information retrieval with the use of the analysis of particular identifiers is described in [17].

There is also an ongoing research for the similarity measures for other parts of the documents, as for example bar charts [18] or images [19], as well as classification and categorization of environment scenes [20].

The main novelty of our anti-plagiarism solution for mathematical documents presented in this paper is based on comparison of verbalized content instead of pure symbolic notation of mathematical formulae. In what follows we show how verbalization improves the accuracy of plagiarism detection in various scenarios.

2. Plagiarism detection in \LaTeX\ documents – tools and methods
\LaTeX\ is a kind of page description language, providing text commands for every part of the document including complex mathematical equations. It is widely used in scientific environment and the majority of Maths students’ reports and theses are encoded in \LaTeX\. Certainly, the essential part of these documents are pure equations and plagiarism detection should be focused at them. In our experiments we consider the documents with source code in \LaTeX\, which is available for the teacher or examination committee reviewing the particular student’s or scientific work.

2.1. Formula separation
The equations described in \LaTeX\ are surrounded by obligatory environment commands. Therefore the extraction of a proper data set for further analysis can be reduced to parsing the source code with respect to all allowed math environments, including the most popular \begin{math} and \begin{displaymath}, as well as dollar and double-dollar signs.

The length of the equation to be taken into further consideration by the anti-plagiarism algorithm should exceed some minimal value. There are many one-letter identifiers used in mathematical and technical documents, which could be indicated as similar to the other short expressions just due to the limited length. The solution is to omit the equations with the description in \LaTeX\ shorter than parameter, which is by default set at 4 characters for our experiments.

2.2. Comparison of formulae pairs
Having particular equations extracted from the source code of the document, we need to compare them with one another. The comparison should avoid high computational cost, because of the amount of data which has to be processed: the document with a typical scientific manuscript may contain a few hundreds of equations, while the thesis obviously much more. Given thousands of documents in the repository which should be checked by anti-plagiarism engine for duplicated or similar equations, the entire process would require several hundred thousands comparisons of the equations pairs.

Taking into account the limitations specified above, we incorporated two main text comparison algorithms into our anti-plagiarism system:

(i) normalized Levenshtein distance,
(ii) weighted cosine similarity measure.
The Levenshtein distance $D_{Le}(\alpha, \beta)$ is the minimum number of equally weighted single-character operations: removals, additions and replacements, required to change string $\alpha$ into string $\beta$. Therefore it can be considered as the measure of difference between two text strings regardless their length $|\alpha|$ and $|\beta|$ [21]. The normalized Levenshtein distance $D_{LeN}(\alpha, \beta)$ given in Eq. 1 additionally places the resulting values in the range 0 – 100%:

$$D_{LeN}(\alpha, \beta) = 100 \times \frac{D_{Le}}{\max (|\alpha|, |\beta|)}$$  \hspace{1cm} (1)

The second text comparison method is the (weighted) cosine similarity measure. First, two numerical vectors $v = [v_1, \ldots, v_n]$ and $w = [w_1, \ldots, w_n]$ with coordinates being the counts of instances of a particular key word in the given strings are generated. Then the cosine of the angle between $v$ and $w$ is calculated by means of the standard dot product of the two vectors as presented in Eq. 2:

$$\cos \angle (v, w) = \frac{v \circ w}{|v| \cdot |w|} = \frac{\sum_{i=1}^{n} v_i \cdot w_i}{\sqrt{\sum_{i=1}^{n} v_i^2} \cdot \sqrt{\sum_{i=1}^{n} w_i^2}}$$  \hspace{1cm} (2)

Values close to 1 show high similarity of the vectors, while values close to 0 indicate big differences between $v$ and $w$.

To increase the quality of the comparison, more important symbols can be emphasized by the introduction of weights. For example, since brackets are commonly used only for ordering the parts of the expression, they should have less impact at the comparison algorithm than root sign or integral sign. In our experiments we have used the highest weights for basic types of mathematical expressions: roots, fractions, integrals, sums, products and limits, medium weights for components, factors, powers and smallest weights for parentheses and brackets. The exact values of the weights were selected heuristically.

Not all of the plagiarism schemes identified in Section 1.2 are \LaTeX - specific, especially the last three seem to be of more general type. For this reason, these kinds of mathematical plagiarism appear to be more difficult to detect by the tool oriented on \LaTeX files. Therefore, parallel to comparison of pure \LaTeX strings representing particular equations, we have introduced verbalization mode. In this mode we assume that spoken versions of the equations can point some similarities not so clearly visible for encoded representation.

The verbalization engine based on the work initially described in [22] and enhanced in [23] is implemented into similarity algorithm. The engine utilizes external definitions made in Lua scripting language. Thanks to that is was possible to provide both English and Polish verbalization modes without much struggle. Certainly, this kind of comparison involves an external scripting language and rather complicated process of the analysis of the equation structure. This, depending on implementation details, can be a time-consuming task, requiring extra processing power or intensive caching, especially for large sets of source \LaTeX documents.

2.3. Similarity computation for the entire document

To compute the similarity measure for the given mathematical document, we calculate similarity of every pair of equations from the candidate and from subsequent members of the repository. The similarity between documents is the ratio between the amount of similar equations to the overall number of equations.

A pair of equations is considered similar if the chosen similarity measure gives result above the limit, which can be adjusted thorough user interface of the system. The value of this limit is set to 95% by default.
The result of the analysis is the list of documents in the repository with the similarity values calculated regarding the candidate. Due to experimental nature of our system we do not filter these results, as different similarity measures and parameters can give slightly different absolute results and the observations of their mutual relations can be valuable.

Additionally, we have introduced one more auxiliary measure, based on the comparison of the entire mathematics in the documents, concatenated into single strings. We discuss the results of this approach in Section 3.2.

![User interface of the test application.](image)

**Figure 1.** User interface of the test application.

### 2.4. Test application

The anti-plagiarism system was developed as a classic desktop application. It was compiled for MS Windows operating system and all the tests were run using Windows 10 Professional. The test application provides simple graphical interface for the parameters setup and the results section (see Figure 1). The operator can choose one input file as the candidate or many files as the repository.

The following parameters can be adjusted:

(i) verbalization mode – additional verbalization of the \LaTeX notation before comparison,
(ii) comparison function – the similarity measure to be used,
(iii) minimal length of the equation to be taken into consideration,
(iv) similarity level needed for the equations pair to be considered as similar in overall statistics,
(v) table switch – an option to show additional formatted data in the end of the report,
(vi) log switch – an option to clear the results log before the actual comparison.

### 3. Experimental results

In order to test our application against recognition of various types of \LaTeX specific plagiarism, a test data set containing a sample reference file and seven modified versions of the reference
file has been generated. Each of the modified files corresponds to exactly one of the plagiarism schemes identified and listed in Section 1.2. Additionally, all test files were compared with some other, not particularly related \LaTeX documents containing many mathematical expressions and formulas, to check the algorithm for coincidental plagiarism detection.

### 3.1. Sample files

The reference .tex file 1 contains a total number of 21 mathematical formulae written in various environments and styles (e.g. in-line, separate one-line, separate two or more lines). Then the seven test files were prepared by modifying the reference file content according to exactly one of the plagiarism schemes. This way we obtained a set of eight test files (the enumeration corresponds to file numbering):

(i) **reference file**

(ii) **equation environment changing**

There are several environment types available for mathematical equations to be exposed in \LaTeX file. This feature can be used for modification of the content of a .tex file without actual modification of the resulting mathematical content. The typical substitutions include \$, \( \ldots \$\), \$\ldots \$\ or \$\$\ldots \$\$. Other types of tested mathematical environments are: \texttt{equation}, \texttt{math}, \texttt{displaymath}, \texttt{array}, \texttt{eqnarray} and \texttt{cases}.

(iii) **spacing adjustment (white spaces, additional gaps)**

There are few ways of spacing adjustment in \LaTeX documents. Except from commands like \texttt{\vspace} and \texttt{\hspace}, one can force additional spacing in mathematical formulae by using \texttt{\quad}, \texttt{\qquad} and \texttt{\ }. 

(iv) **transformation of formulas by application of mathematical identities**

This is a more general, not \LaTeX-specific type of plagiarism, where mathematical content is rephrased into an equivalent form by application of identities and laws. For this reason, this type is difficult to detect by automated algorithms.

(v) **transcribing parts of formulas into text equivalent form**

Similarly to transformation, this type is not \LaTeX-specific and also difficult to detect. However it may be detected after proper verbalization of the mathematical content.

(vi) **reordering expressions or symbols in formula**

This is another non \LaTeX-specific plagiarism scheme, may refer to reordering whole formulas and environments, as well as single symbols within one environment, like for example interchanging \( a^2 + b^2 = c^2 \) by \( b^2 + a^2 = c^2 \).

(vii) **splitting formula into few shorter or merging few formulas into single expression or equation environment**

This applies to splitting or merging mathematical expressions by for example interchanging a single eqnarray environment with the multiple consequent equation environments, or vice versa.

(viii) **replacement of symbols by other symbols**

Also a non \LaTeX-specific plagiarism scheme, may be applied extensively in various types of documents. This includes renaming the used constants or variables, e.g. \( a^2 + b^2 = c^2 \) replaced by \( x^2 + y^2 = z^2 \).

The test files were compared pairwise with each other, i.e. not only the seven modified files were tested against plagiarism with the reference file, but also each of the seven files were checked against all the others.
measure. The most effective similarities detection in majority of plagiarism schemes is observed with itself, with similarity 100%.

Consisting of a reference file and a tested file the size of respective bubble shows the similarity to one of two similarity measures and one of two operation modes used in the tests. For each pair of detector sensitivity to different kinds of plagiarism and make precision metric to be irrelevant (i.e. the ratio of True Positives to Total of Actual Positives). Also, the specific design of experimental document samples was aimed at verification of detector sensitivity to different kinds of plagiarism and make precision metric to be irrelevant in this setting. The results are presented in four bubble charts in Figure 2 – each corresponding to one of two similarity measures and one of two operation modes used in the tests. For each pair consisting of a reference file and a tested file the size of respective bubble shows the similarity of the two files. For reference we included diagonal pairs, representing tests on a file compared with itself, with similarity 100%.

Clearly, the general effectiveness of the algorithm is noticeably better when using cosine measure. The most effective similarities detection in majority of plagiarism schemes is observed.

3.2. Results and Discussion

All tests were run in both implemented modes: the \( \LaTeX \) source mode and the verbalized mode and with default similarity level and minimal equation length settings. Also, in order to determine the efficiency of the discussed similarity measures for detection of specific kinds of plagiarism, the tests were performed for both measures. As the files by design contain exactly the same set of equations, in this test there are no "False Positives" nor "True Negatives" when comparing particular equations, and the calculated overall similarity is the actual recall of the proposed detector at the equation level (i.e. the ratio of True Positives to Total of Actual Positives). Also, the specific design of experimental document samples was aimed at verification of detector sensitivity to different kinds of plagiarism and make precision metric to be irrelevant in this setting. The results are presented in four bubble charts in Figure 2 – each corresponding to one of two similarity measures and one of two operation modes used in the tests. For each pair consisting of a reference file and a tested file the size of respective bubble shows the similarity of the two files. For reference we included diagonal pairs, representing tests on a file compared with itself, with similarity 100%.

Figure 2. File comparison results for (a) \( \LaTeX \) source and cosine measure, (b) \( \LaTeX \) source and Levenshtein measure, (c) verbalized file and cosine measure, (d) verbalized file and Levenshtein measure.

3.2. Results and Discussion

All tests were run in both implemented modes: the \( \LaTeX \) source mode and the verbalized mode and with default similarity level and minimal equation length settings. Also, in order to determine the efficiency of the discussed similarity measures for detection of specific kinds of plagiarism, the tests were performed for both measures. As the files by design contain exactly the same set of equations, in this test there are no "False Positives" nor "True Negatives" when comparing particular equations, and the calculated overall similarity is the actual recall of the proposed detector at the equation level (i.e. the ratio of True Positives to Total of Actual Positives). Also, the specific design of experimental document samples was aimed at verification of detector sensitivity to different kinds of plagiarism and make precision metric to be irrelevant in this setting. The results are presented in four bubble charts in Figure 2 – each corresponding to one of two similarity measures and one of two operation modes used in the tests. For each pair consisting of a reference file and a tested file the size of respective bubble shows the similarity of the two files. For reference we included diagonal pairs, representing tests on a file compared with itself, with similarity 100%.

Clearly, the general effectiveness of the algorithm is noticeably better when using cosine measure. The most effective similarities detection in majority of plagiarism schemes is observed.
for verbalized mathematical content in combination with cosine measure. Only the reference file (5) is poorly recognized as similar to others, but this is not observed to such extent in case of (5) being a test file. As spoken language is more extensive than symbolic, verbalization significantly lengthens the compared strings and the number of constituting words, therefore if they are compared by means of cosine measure, similar or identical formulas contribute more to the similarity score. This tendency is not observed in case of verbalized mode combined with Levenshtein measure, however the algorithm efficiency improved slightly with respect to the Levenshtein - pure LATEX mode.

Table 1. Results of comparison of (1) with non-related files

| File | Overall similarity | Concatenated similarity | Similar equations found |
|------|--------------------|-------------------------|-------------------------|
|      | LATEX source - cosine measure |                         |                         |
| (i)  | 0.00%              | 63.03%                  | 0                       |
| (ii) | 0.00%              | 62.86%                  | 0                       |
| (iii)| 0.00%              | 55.55%                  | 0                       |
| (iv) | 0.00%              | 75.50%                  | 0                       |
| (v)  | 0.00%              | 33.09%                  | 0                       |
|      | LATEX source - Levenshtein measure |                     |                         |
| (i)  | 0.00%              | 16.59%                  | 0                       |
| (ii) | 0.00%              | 26.20%                  | 0                       |
| (iii)| 0.00%              | 9.55%                   | 0                       |
| (iv) | 0.00%              | 22.50%                  | 0                       |
| (v)  | 0.00%              | 19.71%                  | 0                       |
|      | verbalized math - cosine measure |                    |                         |
| (i)  | 47.62%             | 88.70%                  | 10                      |
| (ii) | 23.81%             | 48.46%                  | 5                       |
| (iii)| 9.52%              | 43.95%                  | 2                       |
| (iv) | 19.05%             | 41.95%                  | 4                       |
| (v)  | 33.33%             | 60.78%                  | 7                       |
|      | verbalized math - Levenshtein measure |                |                         |
| (i)  | 0.00%              | 18.06%                  | 0                       |
| (ii) | 0.00%              | 31.95%                  | 0                       |
| (iii)| 0.00%              | 29.37%                  | 0                       |
| (iv) | 0.00%              | 33.56%                  | 0                       |
| (v)  | 0.00%              | 32.39%                  | 0                       |

Another observation is that the algorithm performance is significantly lower for certain types of plagiarism. This especially corresponds to types (5), (7) and (8) (for both measures used) and additionally (3) and (4) when files are compared using Levenshtein measure. Nevertheless, in other cases the overall similarity evaluated by our plagiarism detection algorithm was usually above 70% in case of cosine measure and above 50% in case of Levenshtein measure.

In order to understand the meaning of the above results, we performed additional tests with comparison of the reference file (1) with five non-related mathematical documents, denoted by (i), (ii), (iii), (iv) and (v), containing 109, 24, 36, 26 and 96 mathematical expressions, respectively, among which there were no copies of expressions contained in the reference file (1). The results of the tests are presented in Table 1. For non-related sources, the Levenshtein measure works better for distinguishing the files - especially in verbalized mode, where cosine measured similarity reaches nearly 90%, the Levenshtein similarity is below the level of 20%.
This proves the differences in performance of our algorithm when used with different measures and indicates possible application areas for the two modes and two similarity measures.

4. Summary

The proposed plagiarism-detecting algorithm has proved to be a reliable and efficient tool for recognition of plagiarism in mathematical content of \LaTeX files. The elaborated design and implementation allow for detection of different types of plagiarism with high success rates. Also, the different modes of operation as well as similarity measure settings make the proposed solution a flexible tool, which can be used for verification of mathematical theses or textbooks accordingly to the suspected plagiarism type or, when there are no specific suppositions, to test the file with all modes and analyze the suspiciously high similarity scores if they appear. When used simultaneously, the two measures and two modes bring the information of the nature of detected similarities and recognize the plagiarism occurrences more accurately.

References

[1] Eisa T A E, Salim N and Alzahrani S 2015 Online Information Review 39 383–400
[2] Alzahrani S M, Salim N and Abraham A 2012 IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42 133–149
[3] Meuschke N and Gipp B 2013 International Journal for Educational Integrity 9 50–71
[4] Raghawatetsagal C, Krinke J and Clark D 2018 Empirical Software Engineering 23 2464–2519
[5] Taerungruang S and Aroonmanakun W 2018 GEMA Online@ Journal of Language Studies 18 186–202 URL https://ejournal.ukm.my/gema/article/view/23578/8347
[6] Alzahrani S 2016 J. Comput. Sci 12 1–18
[7] B¨uy¨uktopa¸c O and Acarman T 2020 Man-Machine Interactions 6 ed Grucu A, Czachórs´ki T, Deorowicz S, Harezlak K and Piotrowska A (Cham: Springer International Publishing) pp 125–135 ISBN 978-3-030-31964-9
[8] Oghbaie M and Zanjireh M M 2018 Journal of Big Data 5 52
[9] Sohangir S and Wang D 2017 Journal of Big Data 4 25
[10] Irving R W 2004 University of Glasgow 91–22
[11] Bier A and Sroczynski Z 2018 2018 Federated Conference on Computer Science and Information Systems (FedCSIS) (IEEE) pp 465–469 URL http://dx.doi.org/10.15439/2018F155
[12] Sojka P, Ruzicka M and Novotny V 2018 Proceedings of the 27th ACM International Conference on Information and Knowledge Management (ACM) pp 1923–1926
[13] Isle R M 2018 arXiv preprint arXiv:1801.08439
[14] Meuschke N, Stange V, Schubotz M and Gipp B 2018 Proceedings SIGIR’18: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval
[15] Bhattacharjee D and Dutta S 2013 Procedia Technology 10 715–723
[16] Meuschke N, Schubotz M, Hamborg F, Skopal T and Gipp B 2017 Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (ACM) pp 2211–2214
[17] Schubotz M, Grigorev A, Leich M, Cohl H S, Meuschke N, Gipp B, Youssef A S and Markl V 2016 Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval (ACM) pp 135–144
[18] Al-Dabbagh M M, Salim N, Rehman A, Alkawaz M H, Saba T, Al-Rodhaan M and Al-Dheelaan A 2014 The Scientific World Journal 2014 1–11 URL http://dx.doi.org/10.1155/2014/612787
[19] Meuschke N, Gondek C, Seebacher D, Breitinger C, Keim D and Gipp B 2018 Proc. ACM/IEEE Joint Conf. on Digital Libraries (JCDL)
[20] Petrakis T, Maskeliuñas R, Damasevičius R, Polap D, Woźniak M and Gabryel M 2017 International Joint Conference on Computational Intelligence (Springer) pp 281–303
[21] Navarro G 2001 ACM Comput. Surv. 33 31–88 ISSN 0360-0300 URL http://doi.acm.org/10.1145/375360.375365
[22] Bier A and Sroczynski Z 2015 Proceedings of the Multimedia, Interaction, Design and Innovation MIDI ’15 (New York, NY, USA: ACM) pp 7:1–7:9 ISBN 978-1-4503-3601-7 URL http://doi.acm.org/10.1145/2814464.2814471
[23] Bier A and Sroczynski Z 2019 Multimedia Tools and Applications 78 28089–28110 ISSN 1573-7721 URL https://doi.org/10.1007/s11402-019-07889-3