Methods for Checking Graduate Qualifying Thesis for the Volume of Borrowing

Oleg Ya. Kutsiy[1]*, *Tatiana Yu.* Tsibizova[1], *Ekaterina V.* Shevtsova[1], and *Tatian*a *Yu.* Komkova[1]

[1]Bauman Moscow State Technical University, 2nd Baumanskaya str., 5/1, 105005, Moscow, Russia

Abstract. The article is devoted to the issues of checking a graduate qualifying thesis for the volume of borrowed text and the problems of controlling the circumvention of this check. We considered some variants of widely used methods of verification bypass in MS Word documents, a method of the bypass, because of which a large amount of the unique text will take part in the verification. We analyzed various ways of implementing this method in MS Word documents and considered options for software and manual control of this method.

1 Introduction

Currently, Russia is actively implementing information technologies in all spheres of society through the organization of several national projects, among the most important of which is the national project "Education", adopted in December 2018 and designed to ensure the competitiveness of Russian education and to bring the Russian Federation to the top ten countries in terms of education quality by 2024. One aim of the project is to "create a modern and secure digital educational environment by 2024 that ensures high quality and accessibility of education of all types and levels" and "modernization of professional education" [1, 2, 3]. In the information society context, new opportunities are opening up to improve the efficiency of educational process management, related to the formation of problem-oriented information and communication social spaces [4, 5]. Hence, requirements are created for the organization of the educational process through the use of electronic information and educational environment (EIEE), which is based on various educational resources, including electronic library systems, knowledge control systems [6].

One of these resources is the creation of an electronic library system that checks all final qualifying papers for the volume of borrowed text. This work is carried out using various software tools that have huge databases of text information taken from various sources, including downloadable works by students themselves [7, 8]. The presence of a huge amount of text information, which is regularly updated, makes it possible to say with some confidence that the information about the volume of borrowed text, which is obtained as a result of such verification, is fairly objective and reliable. However, this confidence disappears if you delve into the problem associated with the existence of various methods that make it possible to easily bypass such a check.

* Corresponding author: kutsiy@bmstu.ru

Inserting large volumes of the unique third-party text, often "garbage" that has nothing to do with the work, allows you to increase the volume of the unique text in the work and reduce the percentage of borrowing [9].

2 Analysis and Methods

In graduate qualifying thesis, you may find a variety of "garbage": texts of foreign articles translated into Russian programmatically, texts from works of art, recent blogs, and often any nonsense. Depending on the current volume of borrowed text, the amount of "garbage" can reach about tens to hundreds of thousands of characters. In the practice of inspections, there were works in which the volume of "garbage" reached 500-700 thousand characters. This method is characterized by the fact that there are no repetitive fragments of sufficient length that could be detected using software tools. You can create such "garbage" only using special software tools [10].

Calculating the required amount of "garbage" based on current work data is quite simple:

$$V_G = \frac{V_B}{P_B} 100 - V_T, \quad (1)$$

where V_B – borrowing, characters; V_T – all thesis, characters; P_B – the required percentage of borrowing.

The volume of borrowed text, knowing the percentage of borrowing in the work, is determined as:

$$V_B = V_T \frac{P_{BC}}{100}, \quad (2)$$

where P_{BC}– the current percentage of borrowing.

Thus, if the work contains 100,000 characters and 100% of the borrowing (someone else's work is taken), then to achieve 20% of the borrowing, you must insert 400,000 characters of "garbage" into the work [11, 12]. If you need to slightly reduce the percentage of borrowing, the amount of required "garbage" is not so large. In practice, usually, no one does such calculations, and the amount of "garbage" is inserted with a large margin.

With such volumes of third-party text, authors often have the question of how to hide this text from prying eyes. This is where various text formats in MS Word come to the rescue:

1) text formatting;
2) labels;
3) SmartArt objects;
4) tables;
5) hiding text under figures, labels, graphic shapes, and SmartArt objects.

We should note that in the practice of checking graduation papers, there were papers with a large amount of "garbage" that the authors did not even intend to hide [13].

Depending on how the graduation papers viewing by the compliance supervisor, it may turn out that you cannot hide the "garbage". Especially if the compliance supervisor does not check the work, but silently loads it for verification and is satisfied with the got result, or opens and views the start of work. In this case, the "garbage" is inserted in the middle or at the end of the work. With a large flow of work, the compliance supervisor may not reach the end of such work, even if he is used to flipping through all the work, and if he still gets to the place with "garbage", he can be distracted from this activity by unnecessary conversations, various questions related to the currently visible fragments of text, etc.

With a small amount of necessary "garbage", it can be scattered throughout the work in separate paragraphs, or even better, add to the end of existing paragraphs. Usually, when checking works for compliance with certain requirements, the standard controller pays

attention to the design of titles, the distance between titles and text, the design of paragraphs, and the text at the end of a paragraph may not interest him. You can detect it if only by chance the eye falls on a fragment with "garbage".

Thus you can detect visible "garbage" by carefully scrolling through the work from the first to the last page. Another thing is to detect hidden "garbage". Here you can use the following method: saving a document in txt format and opening the saved text in MS Word. However, this method of verification is time- consuming and with a large flow of documents cannot be implemented in some cases, hiding "garbage" and is not applicable.

Text formatting allows you to hide large amounts of "garbage" by using various parameters available in the font settings [14, 15]. Figure 1 shows a fragment of the text that looks like some kind of dividing line, containing about 84 thousand characters, which are equal to about 19 pages of dense text (without paragraphs and punctuation marks) with a font size of 12 PT and a single line spacing. Just one line with this amount of "garbage" can solve the problem associated with the allowable amount of borrowing.

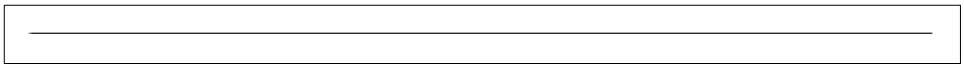

Fig. 1. Text formatting.

You can achieve this effect by setting the minimum font size equal to 1 PT in MS Word, the font scale is 0.5% and all 19 pages of text will turn into a line about 13.5 cm long, which at the width of the text on the page 17 cm will not even reach the right border. You must enter all this text within a single paragraph.

Of course, this line is visible to the naked eye and can lead to unnecessary questions from the person who checks the work. But no one is stopping you from painting it white, making the font transparent, canceling the text fill, or making all the text within this line hidden.

Besides these techniques related to font parameters, you can use the paragraph settings by setting the left margin to 20 cm and this line will disappear from the page. The fact is that the minimum paragraph margins on the left and right, which MS Word allows, are about 55 cm. When using a negative left indent, the text goes beyond the page to the left, and when using a positive left indent and a negative right indent, the text goes to the right outside the page.

You need not use additional formatting options to hide these lines, because you often see framed tables, drawings, and diagrams in your work that are made using MS Word, and no one is stopping you from using such a line in these tables, drawings, or diagrams. Besides, footnotes can be used in works that are separated from the main text by a line, and the resulting line with "garbage" can be used as such a line. The footnote will have to be organized manually.

You can check whether there are workarounds associated with hiding "garbage" using formatting options by carefully viewing the document content after performing the following actions:

1) Select all text in the document using the Ctrl+A key combination. Labels and SmartArt objects will not be included in the selection. They need to be dealt with separately.

2) In the "Font" dialog, set the font size to 14 PT, black text color, on the "Advanced" tab, set the font scale to 100% and the "Normal" spacing between characters.

3) In the "Text Effects Format" dialog on the "Text Fill" tab, set the solid character fill to black and the fill transparency to 0%. This dialog becomes available when you click the "Text Effects..." button in the "Font" dialog.

4) In the "Paragraph" dialog, set the left and right margins equal to 0 cm, the first line indent is 0 cm or 1.25 cm, and the line spacing is 1.5 lines.

This workaround method will be much more reliable if you assign the verification of its use to a program that is engaged in determining the volume of borrowing [16, 17]. If you find numerous characters in a document with invalid formatting parameters, do not allow such a document to be scanned. These parameters include:

1) the font size is less than 8 PT;
2) the color of the symbol is close to the background color on which it is drawn;
3) no color filling of characters;
4) transparency of characters above 50%;
5) the font size is less than 70%;
6) font compaction above 2 PT;
7) the indent in the paragraph on the left or right is less than -0.5 cm, including the indent of the first line;
8) the line spacing is 20% less than the single one.

Next, let's look at the widely used techniques for hiding text using labels. A label is a text box (container) in which you enter a text to place it anywhere on the page, including fields. To get the decals, you must run the command "Label" is "To Draw the Label" on the tab “Insert”. You can also get the label in another way by inserting various geometric shapes available in the "Shapes" list on the same tab. A geometric shape does not become a label until you insert the text into it using the “Add text” command in the context menu for the geometric shape or on the “Format” tab that appears when you select the shape itself.

Labels allow you to hide extra text using the following techniques:

1) hiding "garbage" outside the borders of the label;
2) placing a label with "garbage" outside the page borders;
3) hiding the label.

Hiding "garbage" outside the borders of the label is very simple. The text escapes the lower border easily if it does not fit in the label. To get this effect, you need not adjust the parameters of the label itself, it is enough to make a small label size. In the practice of checking graduation papers, there were inscriptions with "garbage", the size of which was 1×1 mm. If desired, you can reduce the size of the label to zero, the benefit of MS Word allows you to do this. However, such inscriptions will then be difficult to detect even for the author himself (if life forces you to delete "garbage").

To ensure that the text does not run beyond the lower border, there is a checkbox in the label settings "Adjust the size of the figure for the text", which is disabled when creating the label. You can get to this check box by opening the "Shape Format" dialog with the command with the same name in the context menu for the label and enabling the "Label" tab. Usually, this parameter does not interest the authors of documents and therefore does not change, and even the desired text that should be visible sometimes runs away from the label.

Since labels in papers are used in various diagrams and flowcharts, the ability to hide text using labels is usually always present and you need not create them specifically. Text that goes beyond the label becomes invisible to the standard controllers but will be used when checking for the amount of borrowing.

If this simple option of hiding the text outside the label is not suitable (with software control), then you can use the paragraph settings that were discussed earlier to shift the text to the right or left outside the label.

Another technique that has the same application as the previous version is ***to place the label with "garbage" outside the page borders***. Here, the size of the label does not matter because the label itself and the text in it become invisible. You can achieve this effect by placing the label above or below the text using the commands in the text “Wrapping” list on the “Format” tab. For these labels, you can set the label offset relative to the page borders or other elements on the page (margins, paragraphs, lines, characters, etc.) in the range from -55 cm to 55 cm. This feature contains the “Position” tab in the dialog “Markup” that can

display with the command "More layout options" context menu. In this case, you can set the position of the label horizontally and vertically. Negative offsets cause the label to run to the left and up, and positive values cause the label to move to the right and down. When the size of a standard A4 sheet that is used for writing final qualifying papers (21×29.7 cm), the specified range is more than enough to hide the inscription outside the page borders.

Finally, the last option for using labels to hide text is to ***hide the label itself***, which causes the label to become invisible. In MS Word, this is not possible. However, because the docx document is a zip archive (only the file extension is different), which stores a variety of information about the current document in XML files, including the document itself, in the file document.xml, it is possible to directly modify the parameters of the document and its parts. XML files are text files that can be opened in any text editor, including “Notepad”. If the author of the document is a bit of an XML format is an advanced user, and knows where to write a special attribute responsible for the visibility of the label, then he can implement this workaround. We will not describe the implementation of this method, so as not to tire readers.

In the practice of checking graduation papers, this method is used rarely and was seen only among students who are programmers or have studied programming in a large volume.

3 Discussions and results

To control this workaround manually, you can also use the universal method associated with saving the document content in txt format and comparing the saved text with the document text. However, since labels are often used to hide text, you can select all the text in the document by pressing Ctrl+A, set the red color of the text, and view the document for labels. The text in the labels will remain black, and they will be visible against the background of the red text of the document. If there are parts of the document that are designed as labels, such a document can be safely sent for correction, since it is impossible to get labels with the text of the document by accident (as students like to justify themselves), this is done intentionally.

As for checking this workaround method using software control, we should note that the floating position (above or below the text) of the objects in question is not required for registration of graduate qualifying thesis. Therefore, it suffices to monitor the application of settings that lead to such a floating position and force authors to embed these objects in the text, and the use of this method will be excluded [18, 19].

Control of all these methods of circumvention can be implemented programmatically by connecting text identification algorithms [20]. However, these algorithms can only work if there is a limited set of topics for graduation papers at the university or when using "garbage" that includes an obvious discrepancy between the text and specific work. Within a country, such verification is unlikely to be effective.

In figure 2, the green graph shows the value of the indicator associated with the work topic. Small drops in the indicator are associated with a slight difference in the text from those texts that were used for learning the algorithm. A sharp drop in the index maybe because of the text going beyond the topics used in the university, which is possible. Here, the study of the final qualifying work of the student, this is because of "garbage", which in a large enough volume was added in three places in the work. The purple graph shows a significant departure from the subject of the university.

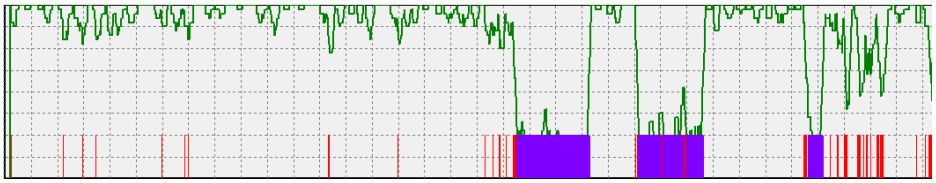

Fig. 2. Significance of indicators related to the work topic.

If necessary, the algorithm for identifying texts can be trained in works within faculties or departments and carry out a narrower control of the subject of works.

4 Conclusion

Thus, the organization of work on the collection and verification of final qualifying works for the amount of borrowing within the university is quite a complex process. The solution is to have an information system that allows you to centrally download works, to see the results of verification promptly, including both the correct design and structure of final qualifying works and to evaluate the content of the work itself, achieving the best results of the quality of training of university graduates.

In conclusion, we would like to note that in this article, the authors have considered some options for implementing and controlling the method of bypassing the verification of documents for the volume of borrowing. We will discuss other workarounds in subsequent articles.

References

1. A.A. Aleksandrov, Fang Ke, A.V. Proletarsky & K.A. Neusypin, *Conception complex continuous education with innovative information technologies*, Proceedings of 2nd International Conference on Education and Education Management (EEM 2012), pp. 374-378 (2012). Retrieved from www.scopus.com
2. E.V. Smirnova, A.A. Dobrjkov, A.P. Karpenko & V.V. Syuzev, *Mentally structured educational technology and engineers preparation quality management*, Communications in Computer and Information Science, 754, pp. 119-132 (2017). Retrieved from www.scopus.com
3. A.A. Aleksandrov, K.A. Neusipin, A.V. Proletarsky & K. Fang, *Innovation development trends of modern management systems of educational organizations*, Proceeding of 2012 International Conference on Information Management, Innovation Management and Industrial Engineering, ICIII 2012, 3, pp. 187-189 (2012). Retrieved from www.scopus.com
4. M.G. Sergeeva, Y.A. Chighovskaya-Nazarova, S.V. Dmitrichenkova, S.Y. Papirovskaya, V. A. Chauzova & I. S. Andryushchenko, E*ffectiveness verification of the application of imitation methods of education in the training of a specialist*, Espacios, 39(38), 11 (2018). Retrieved from www.scopus.com
5. R.Sh. Akhmadieva, L.N. Ignatova, G I. Bolkina, A.A. Soloviev, D.V. Gagloev, M.V. Korotkova, V. I. Burenina, *An attitude of citizens to state control over the internet traffic*, Eurasian Journal of Analytical Chemistry, 13(1), em82 (2018). Retrieved from www.scopus.com
6. N.A. Serdyukova, V.I. Serdyukov, S.S. Neustroev & S.I. Shishkina, *Assessing the reliability of automated knowledge control results*, IEEE Global Engineering Education

Conference, EDUCON, pp. 1425-1428 (2019). doi:10.1109/EDUCON.2019.8725173 Retrieved from www.scopus.com

7. P.H. Ho, T.H. Vo & N.A.T. Nguyen, *Data warehouse designing for Vietnamese textual document-based plagiarism detection system*, 2017 International Conference on System Science and Engineering, ICSSE 2017, 8030873, pp. 239-243 (2017). Retrieved from www.scopus.com
8. A.Y. Gasparyan, B. Nurmashev, B. Seksenbayev, V.I. Trukhachev, E I. Kostyukova & G.D. Kitas, *Plagiarism in the context of education and evolving detection strategies*, Journal of Korean Medical Science, 32(8), pp. 1220-1227 (2017). Retrieved from www.scopus.com
9. M. Ďuračík, E. Kršák, P. Hrkút, *Current Trends in Source Code Analysis, Plagiarism Detection and Issues of Analysis Big Datasets*, Procedia Engineering, 192, pp. 136-141 (2017). Retrieved from www.scopus.com
10. M.N. Halgamuge, *The use and analysis of anti-plagiarism software: Turnitin tool for formative assessment and feedback*, Computer Applications in Engineering Education, 25(6), pp. 895-909 (2017). Retrieved from www.scopus.com
11. A. Akbar & M. Picard, *Understanding plagiarism in Indonesia from the lens of plagiarism policy: Lessons for universities*, International Journal for Educational Integrity, 15(1), p. 7 (2019). Retrieved from www.scopus.com
12. Z. Wang, *Plagiarism in online literature publishing in China: why is it so rampant?*, Online Information Review, 43(4), pp. 551-564 (2019). Retrieved from www.scopus.com
13. A.M. Fazilatfar, S.E. Elhambakhsh & H. Allami, *An Investigation of the Effects of Citation Instruction to Avoid Plagiarism in EFL Academic Writing Assignments*, SAGE Open, 8(2) (2018). Retrieved from www.scopus.com
14. J. Levine & V. Pazdernik, *Evaluation of a four-prong anti-plagiarism program and the incidence of plagiarism: a five-year retrospective study*, Assessment and Evaluation in Higher Education, 43(7), pp. 1094-1105 (2018). Retrieved from www.scopus.com
15. S. Awasthi, *Plagiarism and academic misconduct: A systematic review*, DESIDOC Journal of Library and Information Technology, 39(2), pp. 94-100 (2019). Retrieved from www.scopus.com
16. P.K. Suresh Kumar, *Similarity index of doctoral theses submitted to universities in Kerala: An investigation*, Library Philosophy and Practice, p. 2130 (2019). Retrieved from www.scopus.com
17. Q. Li & C. Zhang, *Research on algorithm of program code similarity detection*, 2017 International Conference on Computer Systems, Electronics and Control, ICCSEC 2017, 8446728, pp. 1289-1292 (2018). Retrieved from www.scopus.com
18. N.I. Martishina, *"Anti-plagiarism" system in self-regulation of scientific activity*, Vysshee Obrazovanie v Rossii, 27(6), pp. 50-57 (2018). Retrieved from www.scopus.com
19. C. Kunschak, *Multiple uses of anti-plagiarism software*, Asian Journal of Applied Linguistics, 5(1), pp. 60-69 (2018). Retrieved from www.scopus.com
20. D.B. Dasari & K.V.G. Rao, *Context similarity strategy for text data plagiarism detection*, International Journal of Engineering and Technology (UAE), 7(2), pp. 14-17 (2018). Retrieved from www.scopus.com