PROGRAM FOR CHECKING THE UNIQUENESS OF THE CONTENTS OF A DOCUMENT "OFFLINE ANTIPLAGIAT"

I. V. Tsapko, Ph.D. in Technology, Assistant Professor,
H. Mukombero, student from group 8I8A.
National Research Tomsk Polytechnic University
E-mail: tsiv@tpu.ru

Introduction

The need to verify the uniqueness of the contents of documents submitted by students as part of their academic work has always been a central problem in universities. The main purpose of this verification is to encourage students to complete assignments on their own, which in turn raises the standards of in-depth research work and the quality of the results presented by students. The most common form of plagiarism checking currently in use is online verification of published materials against global academic databases. However, with regards to the offline checking for plagiarism amongst submitted papers within universities, a few offline systems exist offering the required functionality. This work is aimed at solving this problem and proposes the development of an application for comparing the uniqueness of documents located on a local disk.

Theoretical understanding of plagiarism detection methods

Plagiarism is the use of other people's words or ideas without crediting the source but rather, presenting them as your own. Some of the types of plagiarism include:

- Verbatim plagiarism, also known as copy and paste plagiarism, involves directly copying and pasting text from a source without attribution.
- Patchwork or mosaic plagiarism involves the creation of completely new text by copying phrases and concepts from multiple sources.
- Global plagiarism is when a person completely takes someone else's work and presents it as his own [1].
- Self-plagiarism is reusing previously submitted work or reusing ideas developed from previous assignments with the intent of being credited for it as new material. Although this work belongs to the person, resubmitting it as new material is still considered academic dishonesty seeing as you have already received credit for this work [2].

When comparing student reports, abstracts and other documents located on a local disk, it is of interest to check them for mosaic and global similarity.

Computational methods for similarity detection include approaches such as content similarity detection which is based on the comparison of data fingerprints of different documents, word-for-word comparison of texts, word bag representation of documents, text citation analysis, and stylometry. The performance of these methods strongly depends on the type of plagiarism used. The accuracy of methods based on the search for similarity of texts significantly falls with skillful concealment of borrowings [3].

Plagiarism detection algorithm

Often matching of similar documents is based on counting the maximum number of common words between documents. However, its significant disadvantage is the fact that as the size of the document increases, the number of common words also tends to increase, even in cases where the documents talk about different topics. Cosine similarity and Euclidean distance approaches help overcome this fundamental shortcoming of "general word count". One of the most common algorithms for finding borrowings in various documents is the Shingle algorithm.

A shingle (scale, cell) is a link from which a chain of sentences is built, thereby forming a text. Shingles help to search for individual combinations of words, thereby checking text materials for uniqueness. The main steps of this algorithm [4] are:

- Text normalization (trimming of unnecessary words and punctuation marks).
- Dividing the text into links (the smaller the shingle, the higher the accuracy of the analysis is).
- Comparison of links from different texts.

Based on the above algorithm, a program for checking the similarity/uniqueness of documents was developed in C#. The program allows for the selection of a folder containing the documents for similarity comparison and, after comparing them with the shingles algorithm, gives the similarity result as a percentage.

Testing the prototype of the offline anti-plagiarism program

For convenience of analyzing the functionality of the developed program, 11 test files were placed in the test folder. They contain a systematic combination of 5 sections of text. The texts are symbolically named A, B, C, D and E respectively. For example, a complete combination of texts A, B and C is represented as text 'ABC' (text 1), and a partial combination of texts A, B, C, D and E is represented as text 'abcde' (text 10). The texts were formed as shown in Figure 1.

TEXTS	CONTENTS
0	ABCD
1	ABC
2	AB
3	A
4	B
5	C
6	D
7	E
8	DE
9	ABDE
10	abcde

Fig. 1. Combination of texts.

Upon running the program, the user selects a folder with documents to be compared. After the selection and pressing the "OK" button, a list of all documents stored in this folder appears in the "Texts" list box. When a document is selected, its contents are displayed in the "Contents" text field, and the corresponding percentages of similarity with other documents are displayed in the "Plagiarism" list field, whilst a graphical representation of the comparison result is shown in the graph (Figure 2). The interpretation of the test results of text comparison is shown in Figure 3.

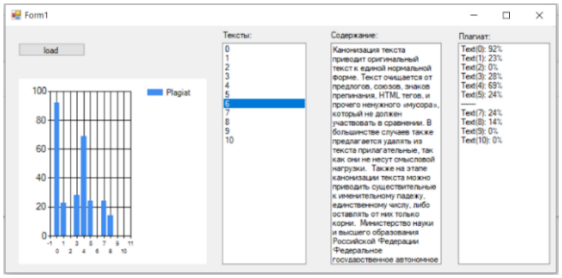

Fig. 2. Graphical interface of the "offline anti-plagiarism" prototype.

TEXTS	CONTENTS \ TEXTS	0	1	2	3	4	5	6	7	8	9	10
0	ABCD		100	96	100	99	100	100	0	30	62	60
1	ABC	82		96	100	99	100	0	0	0	47	55
2	AB	42	51		100	99	0	0	0	0	45	25
3	A	14	17	32		0	0	0	0	0	15	3
4	B	26	32	60	0		0	0	0	0	28	22
5	C	36	44	0	0	0		0	0	0	0	30
6	D	13	0	0	0	0	0		0	28	14	4
7	E	0	0	0	0	0	0	0		66	33	23
8	DE	14	0	0	0	0	0	100	99		50	27
9	ABDE	57	53	96	100	99	0	100	99	100		53
10	abcde	31	35	30	11	44	43	15	39	31	30	

key:

extremely low similarity	0-19
low similarity	20-49
quite similar texts	50-79
extremely similar texts	80-100

Fig. 3. Analysis of the results of the program.

Conclusion

In conclusion, a program was developed that compares the contents of documents and visually shows their similarities. The algorithm used in the program does not fully detect all forms of plagiarism but may be useful when comparing student reports. In the future, the project may be expanded to possibly use machine learning algorithms, create a better graphical interface, and allow the display of plagiarized sections.

List of sources used

1. What is Plagiarism? [Internet]. – // Retrieved from https://www.plagiarism.org/article/what-is-plagiarism (accessed on: 4.01.2022).
2. Bretag, T., & Mahmud, S. A model for determining student plagiarism: Electronic detection and academic judgement. Journal of University Teaching & Learning Practice, 6(1). – 2009 // Retrieved from http://ro.uow.edu.au/jutlp/vol6/iss1/6 (accessed on: 4.01.2022).
3. Classification of Plagiarism Detection Methods [Internet]. – // Retrieved from https://commons.wikimedia.org/wiki/File:Classification_of_Plagiarism_Detection_Methods.svg (accessed on: 15.01.2022).
4. Stein, Benno; Lipka, Nedim; Prettenhofer, Peter (2011), "Intrinsic Plagiarism Analysis" (PDF), Language Resources and Evaluation, 45 (1): 63–82, doi:10.1007/s10579-010-9115-y, ISSN 1574-020X, S2CID 13426762, archived from the original (PDF) on 2 April 2012, (accessed on: 15.12.2021).