УДК 004.08

Т. О. Савчук
Ю. А. Кучевський

УДОСКОНАЛЕНИЙ АЛГОРИТМ ПЕРЕВІРКИ ТЕКСТІВ НА ПОДІБНІСТЬ

Вінницький національний технічний університет

Анотація

Запропоновано алгоритм підвищення якості перевірки текстової інформації на подібність за рахунок використання алгоритму шинглів з можливістю налаштування точності залежно від наявних ресурсів з подальшим порівнянням тексту за допомогою подібності Жаккарда. Це забезпечить швидку та більш якісну обробку великих обсягів текстової інформації.

Ключові слова: плагіат, алгоритм аналізу тексту, семантичний аналіз тексту, інтелектуальний аналіз даних.

Abstract

Provided an algorithm for increasing an effectiveness of checking texts for similarity by using shingle algorithm with ability to set up a precise rank of the algorithm depending on available resources with further text comparison using the Jaccard similarity. It will ensure quick handling of big amount of text information.

Keywords: plagiarism, text analysis algorithm, semantic text analysis, intellectual analysis of data.

Стандартизоване використання алгоритму шинглів приводить до збільшення кількості даних які потрібно обробити, що суттєво обмежує його використання [1]. Стандартний вигляд даного алгоритму передбачає попарне порівняння шинглів кожного документу, і це є його основним недоліком. Нехай n – кількість документів в сховищі даних, m – кількість слів в кожному документів. Візьмемо k за довжину шингла. Відповідно:

$\frac{m}{k}$ – кількість шинглів в одному документі,

$\frac{n \times m}{k}$ – загальна кількість шинглів.

Тоді загальна кількість порівнянь дорівнюватиме:

$$\frac{n \times m^2}{k^2}$$

Зазвичай k є невеликим числом і вкладається в множину Z = {1 … 10}, адже інакше точність алгоритму стане настільки низькою, що подальший аналіз не матиме сенсу [6]. В такому випадку при обчисленні складності алгоритму доцільно вважати k за const. Отже, складність наведеного алгоритму $O(n \times m^2)$, що підтверджує наведений недолік. Затрати пам'яті на зберігання робіт будуть відповідно $O(\frac{n \times m}{k})$ [12].

Для усунення означеного недоліку, пропонується ввести етап попередньої обробки текстової інформації, що додається в сховище даних текстових робіт [2,3,4]. Кожен хешований фрагмент неоднорідності додається в базу даних типу "ключ-значення". Ключем виступає сам фрагмент, а значенням – множина посилань на документи, що містять даний фрагмент. Надалі, під час аналізу текстової інформації на подібність, замість того щоб попарно порівнювати всі шингли поточного файлу, пропонується для кожного шинглу «витягувати» кортеж за відповідним ключем і оновлювати змінні, які зберігають джерела плагіату.Враховуючи те, що сучасні бази даних типу "ключ-значення" гарантують доступ до кортежу за $O(1)$, то після введення описаної оптимізації

складність алгоритму становитиме $O(m)$ порівняно з $O(n \times m^2)$, що є значною перевагою при роботі зі значними обсягами даних [5,6,7,8]. Варто зазначити, що кількість необхідної пам'яті залишається незмінною - $O(\frac{n \times m}{k})$. Після наведених операцій задля точності необхідно обрати найбільші джерела плагіату і порівняти їх з заданим текстом за допомогою схожості Жаккарда [9, 10, 11] використовуючи формулу $JS(A,B) = \frac{|A \cap B|}{|A \cup B|}$.

Отже, удосконалений алгоритм складатиметься з таких етапів:

1. Вибір файлу.
2. Зчитування файлу.
3. Канонізація тексту шляхом приведення до однакового регістру та прибирання пунктуаційних знаків.
4. Розбиття на шингли з заданою довжиною.
5. Обчислення результату хеш-функції за заданим шинглом для кожного фрагменту неоднорідності.
6. Витягнення екземпляру сутності з бази даних за обчисленим ключем.
7. Якщо відповідного запису не знайдено, то створити його. Якщо ж запис знайдено, то дописати ідентифікатор поточної курсової роботи в список за відповідним ключем і оновити змінну з джерелами плагіату.
8. Для найбільших джерел плагіату порахувати схожість Жаккарда та підготувати наочний звіт щодо результату.

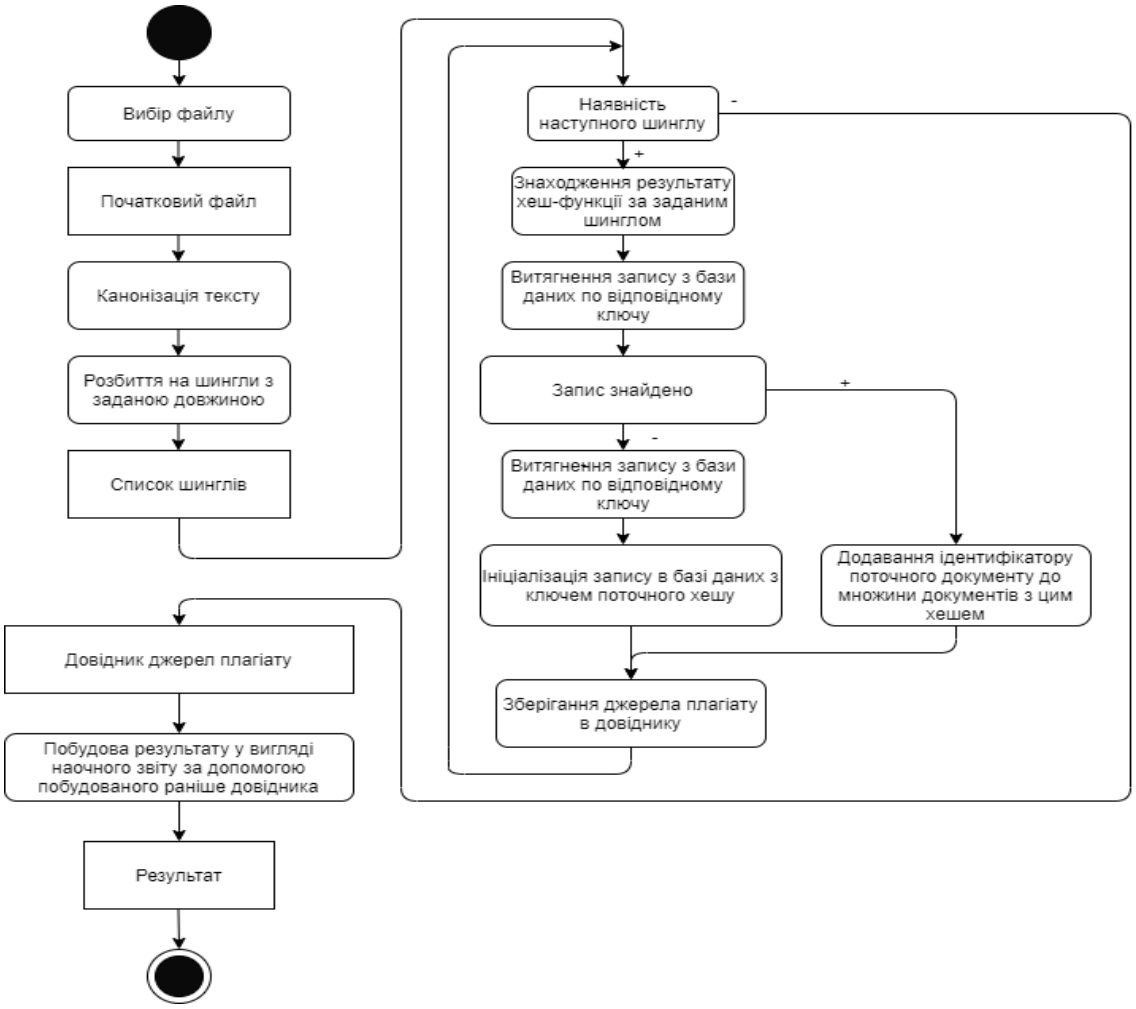

Рисунок 1 – Діаграма активності удосконаленого алгоритму перевірки текстів на подібність

Таким чином, запропоновано удосконалений алгоритм перевірки текстової інформації на подібність, що базується на використанні попереднього зберігання фрагментів неоднорідності в базі даних типу “ключ-значення”. Це забезпечить швидку обробку великих обсягів текстової інформації при незмінному обсязі використаної пам’яті.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Brin S., Davis J., Garcia-Molina H. Copy Detection Mechanisms for Digital Documents — 2001.
2. Monostori K., Zaslavsky A., Schmidt H. Document Overlap Detection System for Distributed Digital Libraries // ACM. — 2000.
3. Meyer zu Eissen S., Stein B. Intrinsic Plagiarism Detection. // Springer. — 2006.
4. Leong A., Lau H., Rynson W. H. Check: A Document Plagiarisment Detection System // ACM. — 1997.
5. Dreher H. Automatic Conceptual Analysis for Plagiarism Detection // Information and Beyond: The Journal of Issues in Informing Science and Information Technology. — 2007.
6. T. O. Savchuk, N. V. Pryimak, A. Assembay, T. Zyska, M. Junisbekov, and A. Annabaev “The technology of searching the associative rules while developing the software”, *Proc. SPIE 10445, Photonics Applications in Astronomy, Communications, Industry, and High Energy Physics Experiments*, 2017, doi: 10.1117/12.2280900.
7. Meyer zu Eissen S., Stein B. Intrinsic Plagiarism Detection. // Springer. — 2006.
8. Седов А. В., Рогов А. А. Анализ неоднородностей в тексте на основе последовательностей частей речи. // Современные проблемы науки и образования.. — 2013. — Вып. 1.
9. Rogers DJ, Tanimoto TT (October 1960). "A Computer Program for Classifying Plants".
10. For example Huihuan Q, Xinyu W, Yangsheng X (2011). Intelligent Surveillance Systems. Springer. p. 161. ISBN 978-94-007-1137-2.
11. Moulton R, Jiang Y (2018). "Maximally Consistent Sampling and the Jaccard Index of Probability Distributions". International Conference on Data Mining, Workshop on High Dimensional Data Mining: 347–356. ISBN 978-1-5386-9159-5.
12. Савчук Т.О., Кучевський Ю.А. Підхід до аналізу на унікальність курсових розробок [Електронний ресурс] – Режим доступу до ресурсу: https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2020/paper/view/8929/7739.

Савчук Тамара Олександрівна — PhD, професор кафедри комп'ютерних наук Вінницький національний технічний університет, м. Вінниця, e-mail: savchtam@gmail.com.

Кучевський Юрій Андрійович — студент кафедри комп'ютерних наук ВНТУ, Вінницький національний технічний університет, м. Вінниця, e-mail: yurii.kuchevskyi@gmail.com

Savchuk Tamara Oleksandrivna — PhD, Professor of the Computer Sciences Chair, Vinnytsia National Technical University, Vinnytsia, e-mail: savchtam@gmail.com.

Yurii A. Kuchevskyi — student of the Computer Sciences Chair, Vinnytsia National Technical University, Vinnytsia, e-mail: yurii.kuchevskyi@gmail.com