УДК 004.08

УДОСКОНАЛЕНИЙ АЛГОРИТМ ПЕРЕВІРКИ ТЕКСТІВ НА УНІКАЛЬНІСТЬ

Савчук Тамара, Кучевський Юрій

Вінницький національний технічний університет

Анотація

Запропоновано алгоритм підвищення ефективності перевірки текстової інформації на унікальність за рахунок використання алгоритму шинглів з можливістю налаштуванням точності, залежно від наявних ресурсів. Це забезпечить швидку обробку великих обсягів текстової інформації.

Abstract

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

Вступ

Стандартне використання алгоритму шинглів приводить до збільщення кількості даних які потрібно обробити, що обмежує його використання [1]. Стандартний вигляд даного алгоритму передбачає попарне порівняння шинглів кожного документу, що є його основним недоліком. Нехай n – кількість документів в базі даних, m – середня 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 не є великим числом і вкладається в множину чисел 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 означений недолік. Затрати пам'яті на зберігання робіт будуть відповідно $O(\frac{n \times m}{k})$.

Для усунення наведеного недоліку, пропонується ввести етап попередньої обробки текстової інформації, що додається в базу даних текстових робіт [2,3,4]. Кожен хешований фрагмент неоднорідності додається в базу даних типу "ключ-значення". Ключем виступає сам фрагмент, а значенням – множина ідентифікаторів документів, які містять даний фрагмент. Надалі, під час аналізу текстової інформації на унікальність, 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 з великими обсягами даних [5,6,7,8]. Варто зазначити, що кількість необхідної пам'яті залишається незмінною - $O(\frac{n \times m}{k})$.

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, ініціалізувавши зі значенням масиву, єдиним елементом якого є ідентифікатор поточного документа. Якщо ж запис знайдено, то дописати ідентифікатор поточної курсової роботи в список за відповідним ключем і оновити змінну з джерелами плагіату.
8. Побудова результату у вигляді наочного звіту за допомогою побудованого раніше довідника.
9. Відображення результату.

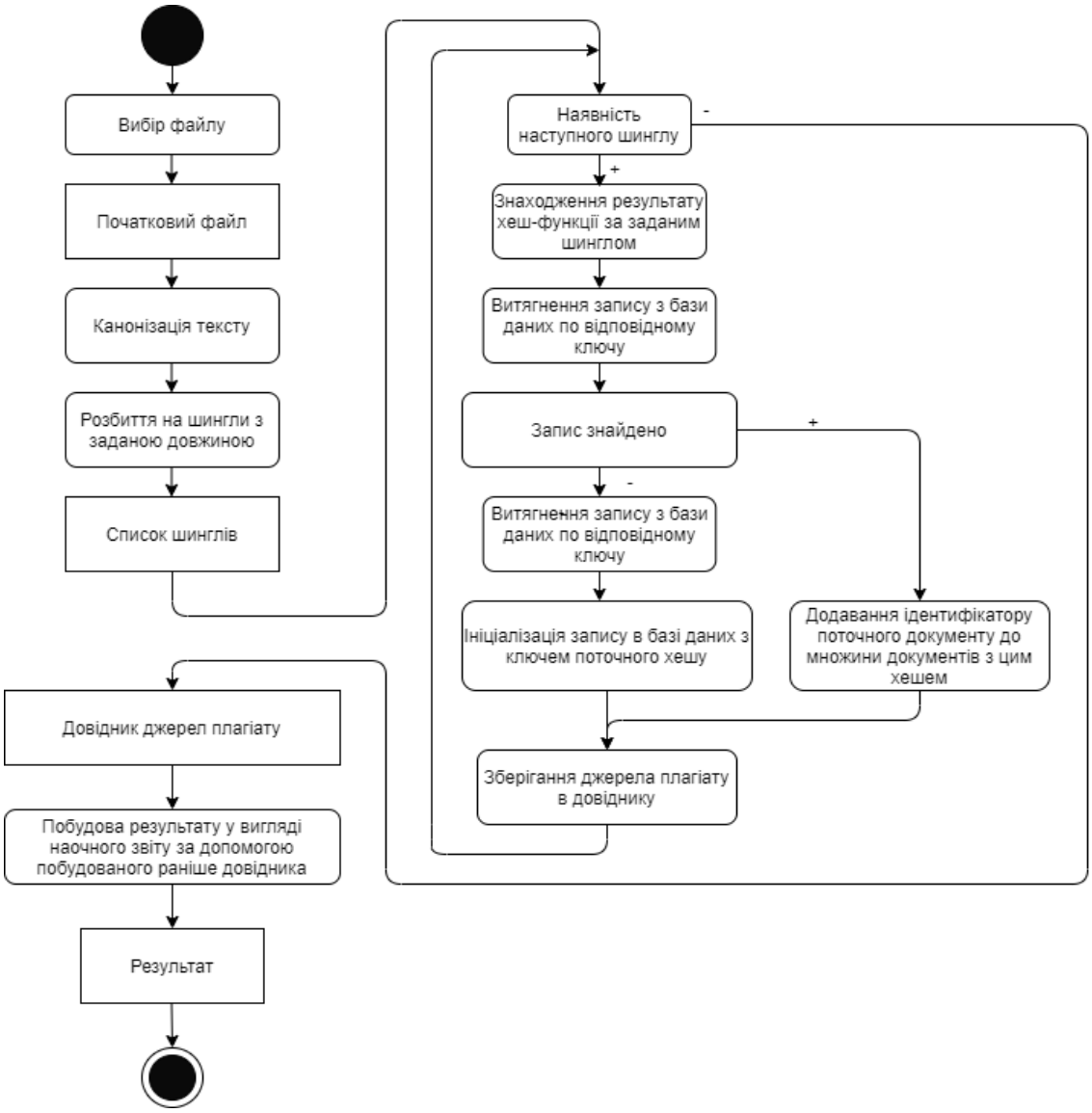

Рисунок 1 – Діаграма активності удосконаленого алгоритму перевірки текстів на унікальність

Таким чином, запропоновано удосконалений алгоритм перевірки текстової інформації на унікальність, що базується на використанні попереднього зберігання фрагментів неоднорідності в базі даних типу "ключ-значення". Це забезпечить швидку обробку великих обсягів текстової інформації. При цьому, кількість використаної пам'яті залишатиметься незмінним.

Доцільність використання удосконаленого алгоритму перевірки текстів на унікальність було підтверджено детальним обчисленням складності алгоритму та подальшим порівнянням зі складністю стандартного алгоритму. Результати проведених досліджень щодо залежності трудомісткості операцій по перевірці текстів на унікальність від обсягу пам'яті яка необхідна для зберігання фрагментів неоднорідності наведені у таблиці 1.

Таблиця 1 – Параметри форматування тексту матеріалів доповіді

К-сть шинглів	**К-сть текстів**	**К-сть операцій (стандарт)**	**К-сть операцій (удосконалений)**	**Затрачена пам'ять (стандарт)**	**Затрачена пам'ять (удосконалений)**
10	10	1000	10	100	100
10	50	25000	10	500	500
50	10	5000	50	500	500
50	50	125000	50	2500	2500
100	100	1000000	100	10000	10000
100	200	4000000	100	20000	20000
200	200	8000000	200	40000	40000

Таким чином, можемо зробити висновок, що кількість операцій виконаних при перевірці тексту на унікальність не впливає на обсяг затраченої пам'яті.

Список використаних джерел

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.