ІНФОРМАТИЗАЦІЯ ВИЩОЇ ОСВІТИ

УДК 004.01

А.О. Білощицький, О.В. Діхтяренко

Київський національний університет будівництва і архітектури, Київ

ЕФЕКТИВНІСТЬ МЕТОДІВ ПОШУКУ ЗБІГІВ У ТЕКСТАХ

Розглянуто кілька методів пошуку збігів у текстах з точки зору їх швидкодії та проведено необхідні експерименти.

Ключові слова: антиплагіат, пошук збігів, метод «шинглів»

Рассмотрено несколько методов поиска совпадений в текстах с точки зрения их скорости и проведены необходимые эксперименты.

Ключевые слова: антиплагиат, поиск совпадений, метод «шинглов»

We consider several methods of searching for a match in the texts in terms of their speed and carried out the necessary experiments.

Keywords: antiplagiat, looking for a match, the method of shingles

Постановка задачі

Для пошуку в тестах однакових фрагментів корисно знати, яким чином ці фрагменти виявилися однаковими. Був це випадковий збіг чи цілеспрямоване копіювання чужої роботи. Але на жаль це неможливо. Тому для перевірки слід розглядати найгірший варіант, коли автор копіює фрагменти чужих робіт та намагається приховати цей факт.

Використовуючи чужі роботи, плагіатор може вдаватися до різних прийомів обману. Ці прийоми можна розділити на дві категорії за призначенням: прийоми обману людини і прийоми обману комп'ютера. Для маскування факту плагіату від людини використовую досить прості прийоми, такі як перестановка слів, абзаців, введення слів, які не несуть змісту, але візуально змінюють текст. Людина, на відміну від комп'ютерної програми, не може тримати в пам'яті повні тексти робіт з теми роботи, що перевіряється, тому може пропустити навіть дослівні копіювання з інших робіт. Але такі маніпуляції досить просто виявляються у процесі комп'ютерного аналізу текстів. Тому, якщо плагіатори знають про можливу перевірку їх роботи комп'ютерною програмою, вони можуть вдаватися до прийомів, спрямованих саме на обман програми. Найпростіший і найлегший варіант – заміна символів кирилиці на аналогічні з латиниці, але цей прийом настільки очевидний, що вже давно не використовується. Також один з можливих варіантів автоматичного приховування плагіату – синонімізація текстів, тобто заміна слів їх синонімами. Для цього існують спеціальні програми-синонімайзери, але їх основний недолік в тому, що програма не аналізує контекст вживання слова, тому тексти часто втрачають початковий зміст. Найбільш реальним на даний момент способом приховання плагіату є «рерайт» тексту. «Рерайт» – це переписування первинного тексту таким чином, що він структурно повністю відрізняється від свого джерела, але передає той самий зміст. Якість «рерайту» може коливатися у дуже широкому діапазоні, в найпростішому варіанті це переставлення місцями слів, використання синонімів, заміна прямої мови на непряму і навпаки. У разі якісного «рерайту» зміст повністю викладається під іншим кутом зору, але маніпулюючи тими ж фактами, що і в джерелі. Такий «рерайт» іноді важко визнати плагіатом, адже треба довести, що при абсолютно різних текстах не додано ніяких нових фактів чи досліджень з теми матеріалу. Істотна відмінність прийому «рерайту» від інших наведених прийомів – це необхідність праці людини, неможливість реалізації автоматичним шляхом.

Зважаючи на вищеописані прийоми, можна сказати, що розпізнати плагіат, якщо він якісно перероблений – майже неможливо. Однак «рерайт» – не такий вже простий спосіб, тому у разі копіювання великих обсягів тексту не виключена можливість того, що певний фрагмент залишиться без змін, або буде змінений досить мало. Тому основна задача програми аналізу – знайти ці фрагменти та передати їх на аналіз експерту.

Аналіз публікацій

Питання плагіату робіт порушувалося вже досить давно. Ще у 1993 році був прийнятий закон України про авторське право та захист суміжних прав [1]. Адже немає нічого простішого, ніж скопіювати чужу, маловідому роботу і видати її за свою. З поширенням доступу до інтернету, швидкість поширення та доступність інформації зросли в рази. Саме пошукові системи зіткнулися з явищем масового копіювання матеріалів і необхідно було знаходити дублікати серед тисяч, а пізніше і мільйонів сайтів. Досить добре описано методи пошуку дублікатів у статті засновників компанії «Яндекс» Ю.Г. Зеленкова та И.В. Сегаловича [2]. Також це питання піднімається на багатьох конференціях, тематика яких так чи інакше пов'язана зі зберіганням чи обробкою даних. Різноманітні варіанти вирішення проблеми пошуку плагіату були описані Ю.В. Кузнєцовим на конференції «Інноваційна модель наукової бібліотеки XXI ст.» [3].

Передумови дослідження

Якість розпізнавання однакових фрагментів напряму залежить від алгоритму пошуку. Сама якість алгоритму може визначатися двома основними мірами – точністю і повнотою.

Точність – це відношення кількості ревелантних елементів до загальної кількості знайдених елементів. У нашому випадку ревелантний елемент – це фрагмент тексту, який дійсно повторюється в двох або більше текстах та має один і той самий зміст.

Повнота – це відношення кількості знайдених ревелантних елементів до загальної кількості таких у тексті. Зрозуміло, що найбільшу точність буде давати пошук по прямих входженнях, а пошук з використанням морфологічного розбору буде давати більшу повноту. Якість пошуку з використанням морфологічного розбору дуже залежить від словників, які при цьому використовуються. Чим більша кількість словоформ у словнику і чим точніше вони описані – тим краще. Але на сьогодні якість словників з української мови, які знаходяться у відкритому доступі, дуже низька. Якість комерційних словників важко перевірити до придбання через обмеження демо-версій доступу. Але крім якості словників, також необхідно звернути увагу і на якість алгоритмів, які проводитимуть пошук в нормалізованому тексті.

Мета статті

Мета статті – проаналізувати різні способи пошуку плагіату в тексті та оцінити їх ефективність. Оскільки, як вже було зазначено, мірами ефективності інформаційного пошуку є точність і повнота, то логічно використовувати саме їх. Але повнота вибірки в даному сенсі буде дуже залежною від способу копіювання тексту. Методи, орієнтовані на знаходження дослівних збігів будуть знаходити тільки їх, а це значить, що якщо в двох текстах буде N дослівних збігів, то всі вони будуть знайдені, і точність методу дорівнюватиме 100%. Якщо ж всі N будуть якось видозмінені, переставлені слова чи ще щось перероблено – метод, орієнтований на точний збіг, дасть повноту в 0%, тобто не знайде нічого. Оскільки ми не можемо спрогнозувати яким чином буде реалізовуватися плагіат в роботах, які ми перевіряємо, то дати належну оцінку алгоритму важко. Також виникає питання, а навіщо нам використовувати методи, які призначені для пошуку точних збігів? Відповідь полягає у тому, що такі методи можуть бути швидшими за методи детального аналізу. У процесі обробки великих масивів інформації швидкість обробки може стати критичною величиною і для комфортної роботи треба буде якось оптимізувати цей процес. Однак швидкість різних способів можна перевірити. Можливо, якщо немає суттєвої різниці у швидкості, можна завжди використовувати алгоритм, який реалізує найглибшу перевірку. Однак мало перевіряти швидкість, треба перевіряти і якість.

Основний матеріал дослідження

У подальших дослідженнях цієї статті перевірялися документи, які свідомо не містили плагіату. Тому для оцінювання «глибини» кожного алгоритму буде використано кількість знайдених збігів. Хоча документи і без плагіату, але вони однієї тематики, тому можливість збігів кількох слів досить велика. При цьому деякі методи, наприклад «шингл», із сортуванням або перехресною перевіркою можуть знайти збіги там, де їх немає. Через це результати цих методів доводиться перевіряти вручну. Теоретично швидкість роботи будь-якого алгоритму залежить від кількості ітерацій в ньому. Але ітерації зовсім нерівноцінні в плані часу виконання, адже на знаходження хеш-суми витрачається більше часу, ніж на перевірку однаковості двох фрагментів тексту. В свою чергу хешування зменшує час перевірки фрагментів, адже

«хеші» перевіряються швидше. Тому теоретично зробити висновок, що буде виконуватися швидше, не можна і необхідно робити експерименти. До часу роботи алгоритму перевірки не входить час канонізації документу за допомогою словників, оскільки ми вважаємо, що ця операція буде проводитися для кожного документу один раз – при імпорті його в базу. Отже, при перевірці текст буде вже канонізований незалежно від обраного способу подальшої перевірки.

Перший розглянутий метод – метод, який реалізований в модулі перевірки плагіату Moodle Crot. Суть методу дуже проста: спочатку із тексту видаляються слова до трьох символів, а потім всі небуквені знаки, тобто дефіси, крапки, пробіли і так далі. У результаті ми отримуємо суцільний ланцюжок букв, який потім з певним кроком n «нарізаємо» на частини по *N* символів у кожній. Від кожної частини беремо хеш-функцію і її результат поміщаємо в набір. Потім два набори різних текстів порівнюємо між собою. Ефективність та швидкість алгоритму залежить від показників *n* та *N*. Оптимальні показники можуть бути різними для різних видів документів, але їх підбір виходить за межі цієї статті. Тому будемо використовувати середньостатистичні оптимальні параметри. Найбільшу точність і відповідно найменшу продуктивність дасть *n*=1. Щоб знайти оптимальний показник *N*, необхідно звернутися до статистики, а також визначитись, яку довжину ланки збігів будемо вважати за плагіат. У документах однакової тематики можуть зустрічатися типові словосполучення по 2-3, а інколи і більше слів, які не можна вважати плагіатом. Але і розраховувати на велику довжину ланки збігів теж не можна, адже автор роботи може приховувати факт плагіату, вставляючи копії маленькими частинами. Тому було прийнято, що плагіатом буде вважатися ланка довжиною приблизно у шість слів. За даними [4] середньостатистична довжина українського слова в інтерв'ю: 5,2 фонеми. Стиль інтерв'ю відрізняється від наукового тим, що він ближче до розмовного і тому спостерігається частіше використання коротких слів. Отже, у наукових текстах середня довжина слова буде більше 5,2 фонем. Враховуючи, що кількість фонем не завжди дорівнює кількості символів у слові, візьмемо за середнє значення шість символів. Тепер можна підрахувати, що наш ланцюжок у шість слів буде в середньому містити 6*6 = 36 символів. Тому беремо параметр *N* такий, що дорівнює 36. Важливо розуміти особливість алгоритму. Якщо у двох текстах будуть збігатися поспіль 35 символів, то метод МС їх не знайде, тобто вірогідність – 0%. Якщо збігатиметься поспіль 36 символів, то за *n*=1 вірогідність, що вони будуть знайдені дорівнюватиме 100%. Але, як показав експеримент, використовувати алгоритм з *n*=1 недоцільно, тому що це забирає дуже багато часу і *n* було прийнято рівним 3. Для даного алгоритму вірогідність того, що буде знайдено невірний фрагмент майже дорівнює нулю, оскільки це метод точної перевірки. Однак існує невелика ймовірність того, що в текстах з'являться подібні ланки внаслідок попереднього опрацювання текстів. У результаті було знайдено кілька груп збігів, які сумарно дорівнюють 91 слову. Середній час пошуку становив 19811 мс.

Принцип алгоритму «шинглів» теж доволі простий. Видаляємо з тексту всі сполучники, короткі слова та небуквені символи, крім пробілів. Далі зі слів складаємо «шингли» – короткі послідовності слів. Чим коротша послідовність – тим більше шанс щось знайти. Оскільки ми вважаємо плагіатом збіг послідовністю у шість слів, коротші послідовності збігів нас не цікавлять. В оригінальній версії алгоритму від кожної послідовності («шингла») береться хеш-сума і створюється набір хеш-сум документа. Потім з цього набору відбирається будь-яким методом, в тому числі і випадково, певна кількість «шинглів», аналогічна операція виконується і з другим документом та його набором. Якщо у відібраній множині співпадає один або більше «шинглів» – документи вважаються подібними. Залежно від розмірів документів може бути різна кількість відібраних «шинглів», а також і різні межі кількості «шинглів», які збіглися, для рішення питання вважати документи подібними чи ні. В даному випадку мета алгоритму – виявити чи схожі два документи в цілому. Але для пошуку плагіату це не підходить, адже тут необхідно не тільки знайти схожість, а й показати що збіглося, де і з чим. Тому буде застосовано дещо інший підхід – для перевірки братимуться всі хеш-суми. Таким чином, кожен «шингл» буде перевірено, також ми можемо відслідкувати який саме «шингл» збігся і показати цей збіг у тексті. За результатами п'яти спроб, середній час виконання перевірки становив 1072 мс, за цей час знайдено 42 слова.

Алгоритм «шинглів» із сортуванням відрізняється лише одним доповненням – сортуванням слів у шинглі перед підрахуванням з них хеш-суми. Якщо звичайний алгоритм міг знаходити лише точні збіг слів і послідовність, то в даному випадку послідовність вже не грає ролі, що розширює межі можливих збігів. Решта алгоритму реалізована так само, як і попередній, для перевірки беруться всі «шингли». За результатами п'яти перевірок середній час виконання алгоритму становив 1095 мс. Сумарно було знайдено збіг 60 слів.

Ще один варіант модифікації алгоритму «шинглів» – перехресна перевірка з допуском збігів. За цим способом в «шингл» поміщають не хеш-суми, а безпосередньо самі ланцюжки слів. Суть методу в тому, що коли ми беремо для перевірки два «шингли» довжиною у шість слів і одне слово в «шинглах» при цьому не збігається – все одно вважається, що «шингли» однакові. При цьому порядок слів у «шинелі» значення також не має. Мінусом цього підходу є велика кількість перевірок на кожній ітерації, але такий спосіб дуже гнучкий. Оскільки ми працюємо вже не з хеш-сумами, а зі словами, то можна використовувати словники, щоб перевірити слово з використанням словників синонімів, антонімів або метрики Левенштайна.

Як тестовий стенд для перевірки роботи алгоритмів використовувався комп'ютер з процесором Intel i5-450M з тактовою частотою 2,4 ГГц. Цей процесор має два фізичних ядра та підтримку чотирьох потоків, тобто можливість обробляти чотири задачі одночасно. Всі алгоритми були реалізовані мовою програмування C# без розпаралелювання задач у один потік, тому максимальне навантаження на процесор під час тестів становило лише 30% від його максимальної продуктивності.

Перевірялося два тексти за тематикою «Електронний документообіг», розмір першого тексту – 10 415 слів або 90 тис. знаків, розмір другого – 12 363 слів або 108 тис. знаків (таблиця).

Таблиця

Результати тестів

Назва методу	***Середній час роботи, мс***	***Кількість знайдених слів, шт.***
Moodle Crot	19811	91
Метод «шинглів»	1072	42
Метод «шинглів» з сортуванням	1095	60
Метод «шинглів» з перехресною перевіркою	86000	82

Висновок

Як алгоритм для швидкої перевірки тексту найбільше підходить метод «шинглів» із сортуванням. Метод з модуля Moodle Crot знайшов більше збігів, але фрагменти складали іноді і 3-4 слова. Це означає, що середня довжина слова у наукових текстах більша, ніж було прийнято. Крім того, цей метод має не найкращі показники часу. Метод «шинглів» хоч і є найбільш гнучким і має найбільше можливостей для розширення, але найповільніший серед чотирьох розглянутих.

Список літератури

1. Про авторське право та суміжних прав: Закон України від 23.12.1993р. [Електронний ресурс]. – Режим доступу: http://zakon1.rada.gov.ua/laws/show/ 3792-12.

2. Зеленков Ю.Г, Сегалович И.В. Сравнительный анализ методов определения нечетких дубликатов для Web-документов [Електронний ресурс]. – Режим доступу: http://download.yandex.ru/company/download /paper_65_v1.pdf.

3. Кузнєцов Ю.В. Засіб виявлення плагіату у наукових текстах: Матеріали міжн. наук. конф. «Інноваційна модель наукової бібліотеки XXI ст. – 2012». – Київ, 2012.

4. Макухіна Т.В., Ліпатов В.М. Особливості фонемної структури українських та англійських текстів інтерв'ю: Матеріали наукової конференції «Наука и технологии: шаг в будуще – 2007». – Дніпропетровськ, 2007.

Стаття надійшла до редколегії 21.06.2013

Рецензент: д-р техн. наук, проф. Ю.М. Тесля, Київський національний університет будівництва і архітектури, Київ.