УДК 519.688

К.т.н., доцент Замятін Д.С., студент Романенко А.О.

**Національний технічний університет України
«Київський політехнічний інститут»**

ВДОСКОНАЛЕННЯ АЛГОРИТМУ ШИНГЛІВ ДЛЯ ЗНИЖЕННЯ ЧУТЛИВОСТІ ДО ПЕРЕСТАНОВКИ СЛІВ

Abstract

Denis S. Zamyatin, assoc. prof., PhD; Artur Romanenko, student

Shingle algorithm improvement for reducing sensitivity to words permutation

This article describes two different approaches for modifying Shingles algorithm in order to improve quality and to make it less sensitive to words permutation. Two distinct approaches for solving this task are proposed. The first one is based on words sorting while the second one is based on using commutative function. The comparative analysis of efficiency of both methods is fulfilled.

Вступ

На сьогоднішній день у глобальному веб-просторі накопичився колосальний об'єм інформаційних джерел. Враховуючи те, що велика частка цієї інформації (приблизно 30%) припадає на копії документів, плагіат та спам, ефективний пошук в інтернеті постає досить складною задачею [1]. Сучасні темпи росту кількості інформації в веб-просторі додатково виставляють високі вимоги до ефективності її обробки.

Проблема пошуку нечітких дублікатів піднімалася ще на початку 90-х років. Одними з перших досліджень в області пошуку нечітких дублікатів є роботи *U. Manber* [2] і *N. Heintze* [3]. Найбільш вживаним серед усіх методів, що були запропоновані до цього часу, можна вважати алгоритм шинглів (від англійського shingle – черепиця).

Алгоритм шинглів був запропонований *А. Бродером* в 1997 році [4]. Основна ідея полягає в розбитті тексту на послідовності слів однакової довжини – шингли. Важливою особливістю є те, що шингли виділяються не один за одним, а накладаються для запобігання втрати інформації.

Класичний алгоритм Шинглів мав ряд недоліків, пов'язаних, в першу чергу, з швидкодією та великими вимогами до пам'яті. Тому пізніше *А. Бродером* було запропоновано схему шинглування і створення короткого образу документу на основі методів «n мінімальних елементів в

перестановці» і «мінімальні елементи в n перестановках» [5]. Крім того, для ряду мов, зокрема української, характерне явище вільного порядку слів у реченні, що може призводити до погіршення результатів роботи алгоритму. Частково проблему чутливості до перестановки слів було розглянуто в статті *Кузнєцова А.Ф.* [6]. Так для зменшення чутливості автор визначав хеш-суму шинглу як суму числових ідентифікаторів кожного слова, що входили до даного шинглу. При цьому значення ідентифікатора окремого слова вибиралося зі словника базових форм слів. Такий підхід дозволив покращити результати, але він мав недолік, пов'язаний з великою кількість колізій.

Постановка задачі

В рамках даної роботи була поставлена задача подальшого вдосконалення алгоритму шинглів, за рахунок створення нових методів зниження чутливості алгоритму до перестановок слів в реченні.

Методи зменшення чутливості алгоритму до перестановки слів

Спочатку сформулюємо задачу наступним чином. Визначимо S_1, як один з шинглів, сформованих для деякого текстового документу, а S_2 - як шингл отриманий з S_1 шляхом зміни місцями двох довільних слів. Тоді можна записати, що

$\boldsymbol{S_1 = w_1,\ldots,w_i,\ldots,w_j,\ldots,w_n}$,

$\boldsymbol{S_2 = w_1,\ldots,w_j,\ldots,w_i,\ldots,w_n}$, де w_i , w_j – переставлені місцями слова.

Нехай SH_1 і SH_2 хеш-суми, обчислені для шинглів S_1 та S_2 за допомогою деякої хеш-функції *hf*. Тоді задача зменшення чутливості до зміни порядку слів зводиться до задоволення наступної умови:

$\boldsymbol{SH_1 = hf(S_1) = hf(S_2) = SH_2}$, або

$\boldsymbol{SH = hf(w_1,\ldots,w_i,\ldots,w_j,\ldots,w_n) = hf(w_1,\ldots,w_j,\ldots,w_i,\ldots,w_n)}$.

Далі буде запропоновано два окремих підходи, до розв’язання цієї задачі.

Сортування слів у шинглі

Суть даного методу полягає в наступному. Під час етапу розбиття канонізованого тексту на шингли, усі слова в межах кожного шинглу додатково сортуються за алфавітом. Це означає, що при будь-якій зміні порядку слів у шинглів, він завжди матиме вигляд відсортованої послідовності:

$\boldsymbol{S = w_а,w_б,w_в,\ldots,w_я}$.

Отже і хеш-суми для двох шинглів S_1 та S_2 (який отриманий з S_1 методом перестановки слів) будуть однаковими, оскільки вони розраховуються для однакових аргументів:

$$S_1 = S_2 = S \Rightarrow SH = hf(S_1) = hf(S_2) = hf(S) .$$

Важливою перевагою даного методу є відсутність будь-яких колізій, пов'язаних з його застосуванням. Тобто хеш-суми двох шинглів будуть рівними тільки в тому випадку, якщо вони складаються з однакового набору слів. Також потрібно відмітити простоту програмної реалізації.

Застосування комутативної функції

Цей метод передбачає використання певної комутативної функції на етапі обчислення хеш-сум шинглів.

Спочатку для кожного слова в межах шинглу обчислюється хеш-сума за допомогою криптографічної хеш-функції hf_w. Після цього для всіх слів шинглу обчислюється значення комутативної функції *cf*, яке і буде вхідним параметром при розрахунку контрольної суми шинглу. А так як результат комутативної функції *cf* не залежить від порядку операндів (в нашому випадку хеш-сум окремих слів), то і значення контрольної суми шинглу також буде незалежним від порядку слів:

$$SH = hf(S) = hf(cf(hf_w(w_1), hf_w(w_2), \ldots, hf_w(w_n))) .$$

Важливим є вибір конкретної комутативної функції, адже це безпосередньо впливає на результати і роботоздатність підходу взагалі. Важливими критеріями є швидкодія, простота в реалізації та мінімізація можливих колізій при обчисленні хеш-сум шинглів. В рамках даного методу найкраще підходить функція «виключне АБО» (XOR), яка задовольняє всім трьом вимогам.

Експеримент проводився для трьох алгоритмів: класичний алгоритм шинглів (№1), модифіковані алгоритми шинглів з сортуванням слів (№2) та з використанням комутативної функції (№3). Оцінювалась кількість розпізнаних дублікатів (*N*) та швидкодія (*Sp*). Виходячи з того, що середня довжина речення в українській мові складає 10–15 слів, довжини шинглів для експерименту бралися з діапазону 5–12 слів. Межа для розпізнання документів як нечітких дублікатів складала 80%.

В якості експериментальної колекції використовувався набір зі 100 документів історичної тематики. 90% з цих документів є нечіткими дублікатами, отриманими з оригіналу шляхом незначних модифікацій та обов'язкової перестановки деяких слів. Об'єм тексту в окремому документі складав 5000 – 15000 символів.

Тестування методів проводилось на наступній платформі: двоядерний процесор AMD Turion 64 X2 TL-58 1.90 GHz, об'єм

оперативної пам’яті 2 Гб, 32-бітна операційна система Windows 7 Professional.

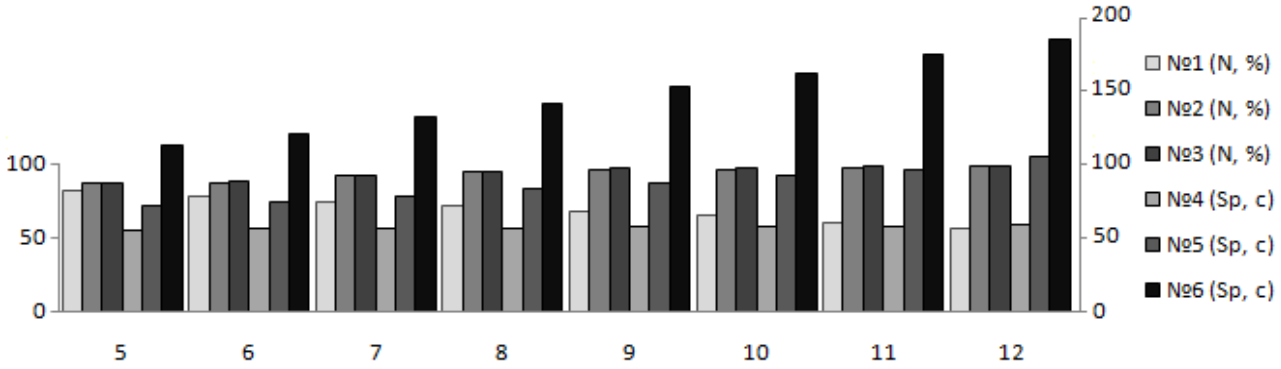

Рис. 1. Результати експерименту

За результатами експерименту можна зробити висновок, що запропоновані методи зниження чутливості помітно покращують результати (5–30%), якщо в дублікатах присутні перестановки слів. Негативним моментом є зниження швидкодії мінімум на 25% в порівнянні з класичним алгоритмом. Особливо це стосується методу №3, що можна пояснити частим обчисленням хеш-сум для кожного слова.

Висновки

В даній роботі запропоновано два методи зниження чутливості алгоритму шинглів до перестановки слів в реченні. При порівнянні запропонованих алгоритмів з класичним можна відмітити зростання кількості знайдених дублікатів на 5–30%. Але одночасно значно знижується й швидкодія, що пов’язано з впровадженням додаткових кроків та операцій. Найкраща довжина шинглу, з точки зору якості, для вищезазначених підходів складає 7-10 слів. Важливим є те, що запропоновані методи можна використовувати в алгоритмах супершинглів, мегашинглів та їх модифікаціях.

Подальші напрями роботи буду стосуватися розробки спеціальної комутативної хеш-фукнції та ефективних методів зниження чутливості алгоритму до заміни слів синонімами.

Література

1. *Д.И. Игнатов, С.О. Кузнецов,* О поиске сходства интернет-документов с помощью частых замкнутых множеств признаков [WWW документ]. URL http://raai.org/resurs/papers/kii-2006/doklad/Ignatov.doc (20 лютого 2011).

2. *U. Manber*. Finding Similar Files in a Large File System // Winter USENIX Technical Conference, 1994. – p. 1-10.
3. *N. Heintze*. Scalable document fingerprinting // In Proc. of the 2nd USENIX Workshop on Electronic Commerce, Nov. 1996. – p. 1-10.
4. *A. Broder, S. Glassman, M. Manasse and G. Zweig.* Syntactic clustering of the Web // Proc. of the 6th International World Wide Web Conference, April 1997. – p. 1-13
5. *A. Broder*, Identifying and Filtering Near-Duplicate Documents // in Proc. Annual Symposium on Combinatorial Pattern Matching, 2000. – p. 1-10.
6. *Кузнецов А.Ф.,* Проблема дублирования страниц и поиска нечетких дубликатов в сайтах по экономической тематике [WWW документ]. URL http://st.free-lance.ru/users/rusl_ir/upload/f_4a97d3d360d0e.doc (20 лютого 2011).