УДК: 004.7 : 004.512

О. А. Тодоріко, Г. А. Добровольський

ВИКОРИСТАННЯ НАБОРУ СИГНАТУР N-ГРАМ ДЛЯ ПОШУКУ ЗА СХОЖІСТЮ

Постановка проблеми. Введена операторами інформаційних систем текстова інформація майже завжди містить орфографічні помилки. Для якісного пошуку та аналізу інформації необхідно їх виявити та виправити. Дану задачу можна вирішити за допомогою пошуку за схожістю у словнику еталонів [1, 2, 3].

Раніше авторами були запропоновані алгоритми словникового пошуку за схожістю з попередньою фільтрацією. Обчислювальні експерименти довели їх високу якість [4]. В цих алгоритмах слово представляється як набір бінарних векторів, у яких більшість елементів дорівнює нулю, що робить неефективним використання оперативної пам'яті.

Метою роботи є розробка та дослідження способу пошуку за схожістю у словнику, який вимагає меншого об'єму пам'яті і зберігає всі переваги раніше розроблених алгоритмів.

Аналіз публікацій. Існують різні методи пошуку за схожістю. Найуживанішими є послідовний пошук, метод розширення вибірки або метод спел-чекера, різні типи дерев: trie-дерева, триангуляційні дерева, частотні дерева, kd-дерева, методи n-грам, хешування за сигнатурою [3, 5, 6].

Мірою схожості між словами найчастіше є одна із модифікацій метрики Левенштейна – відстань редагування [3, 7]. Вона визначається, як мінімальна кількість операцій вставки, заміни, видалення символу і перестановки сусідніх символів, необхідних для перетворення одного слова в інше. Таким чином формалізується інтуїтивне поняття про «помилку». Існує багато алгоритмів ефективного обчислення відстані редагування, проте значна обчислювальна складність робить неефективним їх використання для пошуку у великих масивах з використанням повного перебору, чи послідовного пошуку [4, 5].

У методі розширення вибірки [8] до шуканого слова застосовують різні операції редагування і намагаються знайти у словнику змінений варіант, спираючись на відомі способи пошуку точної відповідності. Однак при допущенні 2-х і більше помилок розширена вибірка стає занадто великою. Побудова різних типів дерев [5, 7] дозволяє за рахунок складної структури даних зменшити кількість порівнянь. До недоліків таких алгоритмів можна віднести складність побудови їх добре збалансованих паралельних аналогів.

Інший спосіб прискорити пошук за схожістю – попередня фільтрація за допомогою швидких і неточних функцій порівняння. Найвідомішими з таких методів є методи n-грам і хешування за сигнатурою [5]. Вони легко адаптуються для паралельних обчислень і оптимальні при обробці великих об'ємів даних. Розглянемо їх детальніше.

Метод n-грам [3] використовує припущення, що схожі слова мають спільні підрядки довжини n (n-грами). В інвертованому пошуковому індексі для кожної n-грами зберігається множина слів, що містять цю n-граму. Слід зауважити, що схожі слова з довжиною меншою ніж $2n$ можуть не мати жодної спільної n-грами. В результаті повнота пошуку зменшується.

У методі хешування за сигнатурою [5] для кожного слова v при заданому алфавіті Abc обчислюється сигнатура $signat(v)$ – бітовий вектор розмірності m, k-й елемент якого дорівнює одиниці, якщо в слові v є k-й символ алфавіту. Бітовий вектор $signat(v)$ вважається двійковим записом числа, яке є значенням хеш-функції $H(v)$, що дозволяє організувати словник у вигляді хеш-таблиці. Якщо слово v отримано з u у результаті однієї операції редагування, то в силу визначення сигнатури, бітові вектори $signat(v)$ і $signat(u)$ відрізняються не більше ніж у двох розрядах. Ця властивість дозволяє використовувати сигнатури для пошуку за схожістю.

В даній роботі показано, що комбінація методів методи n-грам і хешування за сигнатурою дозволяє виконати ефективну попередню фільтрацію записів під час пошуку за схожістю та дає максимальну повноту результатів пошуку.

Основна частина. Розглянемо задачу пошуку за схожістю.

Задається словник S, що є множиною з N записів; u – запис словника, що містить слово і, можливо, іншу інформацію про слово, наприклад правила словозміни.

$$S=\{u\},\ |S|=N,\ N>0 \qquad (1)$$

Пошуковий зразок h може відрізнятися від слова запису наявністю зайвих, пропущених, змінених або переставлених символів. До початку пошуку задається точна міра схожості $edist(u, h)$ та її граничне значення d. Наприклад, точною мірою схожості може бути метрика Дамерау–Левенштейна, а її граничним значенням – максимальна кількість помилок d. Результатом пошуку за схожістю є підмножина S' записів словника:

$$S' = \{u : edist(u, h) \le d,\ \forall u \in S'\},\ S' \subset S \tag{2}$$

Пошук виконується у два етапи. Спочатку записи словника перевіряються за допомогою деякої простої приблизної міри схожості $Rel(u, h)$, і знаходиться множина S'':

$$S'' = \{u : \left|Rel(u, h) - extr\right| \le \delta,\ \forall u \in S''\},\ S'' \subset S, \tag{3}$$

де $extr$ – екстремальне значення $Rel(u, h)$; δ задає розмір області навколо екстремуму. Областю визначення приблизної міри схожості є словник та пошукові зразки.

Точна міра схожості обчислюється на другому етапі лише для слів, які належать множині S''. Якісна функція $Rel(u, h)$ повинна мінімізувати потужність множини S'' і водночас обчислюватися набагато швидше, ніж точна міра схожості.

Таким чином на першому етапі відкидаються несхожі на пошуковий зразок записи, що приводить до зменшення вибірки, та заощаджує обчислювальні ресурси.

Розглянемо приблизну міру схожості, яка спирається на представлення слова у вигляді набору сигнатур, побудованому так, що представлення схожих слів є схожими.

Для обчислення сигнатур задається алфавіт $Abc = \{ a_1, a_2, \ldots, a_m \}$, де $|Abc|=m>0$, m – потужність впорядкованої множини Abc. Будь-яке слово словника S є послідовністю символів алфавіту Abc. Сигнатурою $signat(u)$ слова u є бітовий вектор розмірності m, i-й елемент якого дорівнює одиниці, якщо слово u містить i-й символ алфавіту. Отриманий таким чином вектор можна вважати двійковим записом числа – значення хеш-функції.

Для побудови набору сигнатур, слово u представляється у вигляді послідовності символів алфавіту Abc:

$$u=< a_1, a_2,., a_k >,\ a_j \in Abc,\ j=1..k,\ |u|=k,\ k>0 \tag{4}$$

Далі будується частково впорядкована множина кортежів $v = < v_0, v_1, ., v_t>$. Кортеж v_0 містить всі символи слова u. Інші кортежі v_i $(i=1..k-n+1)$ описують його частини, кортеж v_i складається із n символів, які йдуть у слові u підряд, починаючи з i-ї позиції:

$$\begin{gathered} v_0 = < a_1,\ a_2, .,\ a_k >,\ v_1=<a_1,\ a_2,\ \ldots, a_n>,\ v_2=< a_2,\ a_3, \ldots,\ a_{n+1}>,\ \ldots, \\ v_{k-n+1} = <a_{k-n+1},\ a_{k-n}, \ldots,\ a_k >,\ a_j \in Abc,\ j=1..k,\ |v_i|=3,\ i=1..\ k-n+1 \end{gathered} \tag{5}$$

Якщо довжина слова менша величини n, то створюється лише один кортеж v_1, ідентичний кортежу v_0.

На останньому кроці набір кортежів відображається у множину сигнатур $H(u)=\{H_0, H_1, \ldots, H_{t-1}, H_t\}$, де $H_j=signat(v_j)$, $j=1..t-1$, $H_t=|u|=k$, $t=k-n$. Отримана таким чином множина сигнатур $H(u)$ є представленням слова u.

Варто звернути увагу, що сигнатура H_0 будується та використовується особливим чином. Кількість сигнатур може бути меншою, ніж кількість кортежів. На відміну від досліджених раніше варіантів [4] порядок сигнатур $H_1, \ldots, H_{t-1}$ у представленні слова не впливає на результати пошуку за схожістю, що дає можливість зберігати лише ненульові значення. Крім того, множина значень для сигнатур $H_1, \ldots, H_{t-1}$ містить всього M елементів.

$$M = C_m^1 + C_m^2 + \ldots + C_m^n \tag{6}$$

Наприклад, для українського та російського алфавітів ($m=33$) та $n=3$ кількість різних сигнатур $M=6017$, що значно менше кількості усіх можливих n-грам і дозволяє використати для зберігання кожної із сигнатур $H_1, \ldots, H_t$ всього 12 біт. Для порівняння, кількість різних n-грам для n=3 та m=33 дорівнює m^n = 35 937, що потребує 16 біт в індексі на кожну із n-грам.

Побудоване таким чином представлення слів дає можливість приблизно оцінити мінімальне значення відстані редагування між двома словами:

$$Rel(u,\ h) = max(dw,\ dl,\ w) \rightarrow min$$
$$dw = (1/2)\ bitcount(H_0(u)\ XOR\ H_0(h))$$
$$dl = |\ l_u - l_h\ |$$
$$w = \begin{cases} \frac{1}{n}\sum_{i=1}^{l_u} w_i,\ w_i = \min\limits_{j=1..l_h} bitcount(H_i(u)XOR\ H_j(h)),\ якщо\ |u| > |h| \\ \frac{1}{n}\sum_{j=1}^{l_h} w_j,\ w_j = \min\limits_{i=1..l_u} bitcount(H_j(h)XOR\ H_i(u)),\ якщо\ |u| \le |h| \end{cases} \quad (7)$$

де l_u l_h – довжини слів u та h; функція $bitcount()$ обчислює кількість одиниць у двійковому записі числа; XOR – побітове виключне АБО.

У формулі (7) величина dl є різницею довжин слів u та h, dw дорівнює кількості символів алфавіту, які зустрічаються лише в одному із двох слів. Величина w показує мінімальну можливу різницю між множинами сигнатур n-грам. Множник $(1/n)$ з'явився тому, що одна елементарна операція редагування в слові змінює n сигнатур. Величина $Rel(u, h)$ є невід'ємною і для ідентичних слів дорівнює нулю.

Перевагою формули (7) є незалежність її складових, внаслідок чого вони можуть обчислюватися в будь-якому порядку. Для порівняння слід зауважити, що алгоритм знаходження відстані редагування [3, 5] вимагає строго послідовного виконання, тому що на n-му кроці завжди потрібно знати результати $(n-1)$-го кроку, що значно збільшує час обчислення.

Задана таким чином приблизна міра схожості використовується для попередньої фільтрації записів словника – знаходження множини слів, на якій значення Rel є близьким до мінімального.

$$Rel(u,\ h) = max(dw,\ dl,\ w) \rightarrow min \quad (8)$$

З формули (7) легко помітити, що умова фільтрації $Rel(u,\ h) \le d$ завжди зводиться до однієї з умов: $dw \le d$, $dl \le d$, $w \le d$. Завдяки цьому якщо при обчисленні приблизної міри схожості хоча б одна з величин dw, dl або w виявляється більшою, ніж максимально дозволена відстань редагування d, то слова u та h вважаються несхожими. Таким чином, починаючи обчислення по формулі (7) із найпростіших елементів dw, dl можна прискорити фільтрацію.

Оцінка якості представленого алгоритму. Результатом роботи описаного вище алгоритму є впорядкований набір записів, які можуть виявитися схожими на пошуковий зразок. Адекватними чисельними характеристиками якості пошуку в цьому випадку служать макроусереднена середня точність і ефективність відбору індексу [3, 4].

Макроусереднена середня точність (Mean Average Precision – MAP) [3, 4] – це усереднений для різних пошукових запитів за кількістю знайдених записів відсоток релевантних слів в результатах пошуку.

Ефективність відбору індексу – це відсоток слів, які можуть виявитися релевантними [3, 4]. Таким чином, чим більше частка правильно відкинутих слів, тим менше показник ефективності та тим якісніше працює алгоритм. В даному випадку він обчислюється як відношення потужності отриманої після попередньої фільтрації множини S'' до кількості записів словника S.

Тестування проводилося на об'єднаних в одну множину словах з орфографічного словника під ред. проф. Лопатіна, словника російської літератури (http://www.serann.ru/vocabuli/) і всіх слів романів Льва Толстого «Війна і мир» і «Анна Кареніна». Складений таким чином словник містив близько 400 000 слів.

Пошуковими зразками були слова з типовими помилками [4]. Такий набір зразків знаходиться ближче до реальних завдань пошуку і тому має перевагу перед автоматично створеним.

В таблиці 1 показані характеристики якості пошуку за схожістю для запропонованого у даній роботі алгоритму та для аналогічних методів, представлених раніше [1, 2, 4, 6]. Як можна побачити, запропонований у роботі метод має найкращий показник точності MAP та одну із найкращих ефективність відбору індексу.

Таблиця 1 Числові характеристики якості пошуку за схожістю у словнику, який містить 400 тис. слів при максимальній відстані редагування d=2.

Методи	MAP	Ефективність відбору індексу
Варіант 1	0.0505	0.071
Варіант 2	0.0832	0.071
Варіант 3	0.0730	0.071
Варіант 4	0.2152	0.003
хеш. сигнат.	0.2209	0.074
n-грами(*n*=3)	0.0125	0.046
Сигнатури *n*-грам (*n*=3)	**0.3535**	**0.005**

Висновки. Запропонований у роботі алгоритм є розвитком описаних у попередніх роботах методів пошуку за схожістю. Так само, як і попередні варіанти, він стійкий до вставки, заміни, видалення символу, перестановки символів. У процесі пошуку записи, які не задовольняють умовам пошуку, відкидаються, і точна відстань редагування не обчислюється. Таким чином, відбувається звуження вибірки вже в процесі обробки, що дозволяє економити обчислювальні ресурси.

Головна особливість запропонованого алгоритму у тому, що він враховує лише відносні позиції символів, обчислюючи сигнатури *n*-грам. Велика кількість сигнатур робить ефективність відбору індексу дуже високою – правильно відкидаються 99,5% явно несхожих на зразок слів, і, на відміну від класичного методу *n*-грам, помилково відкинутих слів немає. Одночасно з цим мала множина значень для сигнатур *n*-грам та зберігання змінної кількості сигнатур зменшує об'єм пошукового індексу та, відповідно, час пошуку. Кількісні характеристики – макроусереднена середня точність і ефективність відбору індексу, – показали високу якість попереднього пошуку за схожістю.

В даній роботі основна увага приділялась визначенню якості запропонованого алгоритму, і деякі додаткові можливості залишились недослідженими. Наприклад, відображення слова у множину сигнатур робить можливою побудову інвертованого пошукового індексу, що дозволить застосувати відомі [3] методи оптимізації та прискорення пошуку.

ЛІТЕРАТУРА

1. Тодоріко, О.О. Словниковий пошук за схожістю за допомогою хешів на основі сигнатур [Текст] / О.О. Тодоріко, Г.А. Добровольський // Вісник ХНТУ. – № 3(39). – Херсон: ХНТУ. – 2010. – С. 467-471.
2. Тодорико, О.О. Использование хеширования по нескольким сигнатурам для очистки и объединения словарей данных на примере названий географических объектов [Текст] / О.О. Тодорико, Г.А. Добровольский // Весник ХНТУ. – № 3(42). – Херсон: ХНТУ. – 2011. – С. 419-423.
3. Маннинг, Кристофер. Введение в информационный поиск [Текст] : пер. с англ. / Кристофер Д. Маннинг, Прабхакар Рагхаван, Хайнрих Шютце. – М. : Вильямс, 2011. – 528с.: ил. – парал. тит. англ. – ISBN 978-5-8459-1623-5.
4. Тодоріко, О.О. Оцінка сигнатурних алгоритмів пошуку за схожістю в словнику [Текст] / О.О. Тодоріко, Г.А. Добровольський // Вісник ХНТУ. – № 2(41). – Херсон: ХНТУ. – 2011. – С. 250-254.
5. Boytsov, Leonid. Indexing methods for approximate dictionary searching: Comparative analysis [Text] / Leonid Boytsov // ACM Journal of Experimental Algorithmics. – May 2011. – Volume 16, Issue 1. – P.1-91.

6. Тодорико, О.О. Побудова моделі математичного представлення текстових даних, для знаходження рядків схожих за написанням [Текст] / О.О. Тодорико // Вісник ЗНУ. Фізико-математичні науки. – №1. – Запоріжжя: ЗНУ. – 2011. – С. 118-127.
7. Кнут, Дональд Э. Искусство программирования. – Т. 3 : сортировка и поиск – 2-е издание [Текст] / Дональд Э. Кнут. – М.: Вильямс, 2011. – 824 с.
8. Kuenning, G. International spell: a fast screen-oriented spelling checker / G. Kuenning, R. E. Gorin, P. Willisson, W. Buehring, and K. Stevens. – 1988. – Access mode : http://www.lasr.cs.ucla.edu/geoff/ispell.html

ТОДОРІКО Ольга Олексіївна – асистент кафедри інформаційних технологій Запорізького національного університету.

Наукові інтереси:

- методи та технології інформаційного пошуку.

ДОБРОВОЛЬСЬКИЙ Геннадій Анатолійович - зав. лаб. веб-технологій та дистанційного навчання ЗНУ

Наукові інтереси:

− інформаційні технології в освіті, розробка архітектури сучасного програмного забезпечення.