УДК 004.021

Катерина КУЗЬМА
katushke2017@gmail.com
ORCID: 0000-0002-0937-7299
м. Миколаїв

ОБЧИСЛЮВАЛЬНА ТЕХНОЛОГІЯ НЕЧІТКОГО ПОШУКУ В ТЕКСТІ АБО СЛОВНИКУ

В роботі розглянуто обчислювальну технологію нечіткого пошуку, реалізовану для СУБД MySQL з використанням функцій PHP. Запропонована технологія базується на обчисленні відстані Дамерау-Левенштейна. Проаналізовано алгоритм Левенштейна, розглянуто поняття метрики схожості рядків.

Визначено, що обчислювальна технологія ефективно працює під час порівняння пошукового запиту з невеликим за розміром словником. Її доречно застосовувати для пошуку назв та імен за наявним словником, при якому висока ймовірність того, що користувач може зробити помилку набору чи помилитися хоча б на один символ. В результаті проведеного дослідження встановлено, що оскільки на сьогодні MySQL не пропонує вбудованого рішення для реалізації нечіткого пошуку в тексті або словнику, актуальними є питання розробки ефективних обчислювальних технологій, здатних розширити функціональність пошукових систем веб-сайтів.

***Ключові слова**: нечіткий пошук, відстань Дамерау-Левенштейна, обчислювальна технологія, схожість рядків, MySQL.*

Постановка проблеми

Пошук – найпростіший спосіб доступу до текстових даних. Технологія нечіткого пошуку інформації дозволяє розширювати запит близькими за написанням словами, що містяться в будь-якій базі даних, архіві документів, наприклад, електронної бібліотеки.

Сучасні пошукові системи мають такі можливості: індексування тексту; пошук за ключовими словами; морфологічний пошук – пошук за словоформами; логічна мова запитів, яка дозволяє задавати умови спільного входження (не входження) ключових слів у документ; рангування документів відповідно до ключового запиту.

Однак, при сьогоднішніх швидкостях зростання інформації в Інтернет цих можливостей вже не вистачає. Тому сучасні пошукові системи оснащуються додатковими засобами пошуку.

Алгоритми нечіткого пошуку (відомі як пошук за подібністю або fuzzy string search) є основою систем перевірки орфографії та повноцінних пошукових систем таких як Google або Yandex.

Завдання нечіткого пошуку формулюється наступним чином: «За заданим словом знайти в тексті або словнику розміру всі слова, що збігаються з цим словом (або починаються з цього слова) з урахуванням можливих відмінностей». Ця функція має назву «знайти схоже». Наприклад, запит «мультимедіа» може бути розширений словами: «сультимедіа», «іультимедіа», «мультиседіа» тощо.

Область застосування алгоритмів нечіткого пошуку в мережі Інтернет – пошук назв та імен за наявним словником, при якому висока ймовірність того, що користувач може зробити помилку набору чи помилитися хоча б на один символ.

Аналіз останніх досліджень і публікацій

Створенням, дослідженнями алгоритмів нечіткого пошуку займаються Фролов О.С., Желудков А. В., Макаров Д. В., Фадєєв П. В., Харитоненков А.В., Ліманова Н.І., Міняйло А.Ю., Турчина В.А. [1-5]. В більшості робіт зазначених авторів здійснюється порівняння алгоритмів нечіткого пошуку, визначення їх ефективності.

Аналіз робіт [1-5] показав актуальність розробки обчислювальних технологій на базі алгоритмів нечіткого пошуку з метою їх впровадження в інформаційні системи, в яких відсутні вбудовані рішення.

Постановка завдання

Метою роботи є розробка обчислювальної технології нечіткого пошуку для інформаційних систем, реалізованих на СУБД MySQL та її реалізація засобами PHP.

Виклад основного матеріалу

Алгоритми нечіткого пошуку характеризуються метрикою – функцією відстані між двома словами, що дозволяє оцінити ступінь їх подібності в даному контексті.

Функція *d(x,y)* для обчислення відстані між двома векторами *x* та *y* повинна мати такі властивості:

– невід'ємність: $d(x,y)\geq 0 \ \forall x,y$;

– властивість нуля: $d(x,y) = 0 \Leftrightarrow x = y$;

– симетричність: $d(x,y) = d(y,x) \ \forall x,y$;

– нерівність трикутника: $d(x,z)\leq d(x,y) + d(y,z) \ \forall x,y,z$.

Відповідно до наведених властивостей існує можливість побудувати багато різних метрик, однією, з яких є Евклідова метрика:

$$d(x,y) = \sqrt{\sum_i (x_i - y_i)^2} \ . \qquad (1)$$

Проте для завдання обробки текстової інформації така метрика є не досить зручною. Між тим, в більшості випадків під метрикою розглядається загальне поняття, яке не вимагає виконання умови (1). Це поняття називається відстанню (або у більш загальному плані – функцією схожості рядків). Вдало підібрана функція схожості слів враховує різні типи змін у слові, включаючи видалення, заміни, вставки та транспозиції символів, а в найкращому випадку й схожість звучання слів.

Серед найвідоміших метрик – відстані Хеммінга, Левенштейна та Дамерау-Левенштейна. При цьому відстань Хеммінга є метрикою лише на безлічі слів однієї довжини, що обмежує її область застосування.

Обчислювальну технологію нечіткого пошуку слова (словосполучення) в БД MySQL з використанням функцій PHP пропонується реалізувати наступним чином:

1. Обчислити метафон пошукового запиту (в PHP представлений функцією `metaphone()`): індексування слова за фонетичним принципом. Дана функція застосовується тільки для слів англійського алфавіту, тому пошуковий запит кирилицею транслітерується у латиницю функцією `str_replace()`.

2. Знайти всі слова в словнику БД за метафоном з відстанню Левенштейна (або Дамерау-Левенштейна) меншою за два символи.

3. Якщо нічого не знайдено – користувач зробив занадто багато помилок в слові, припиняємо пошук та повідомляємо, що нічого не знайдено.

4. Якщо знайдено одне слово – повертаємо його.

5. Якщо знайдено більше одного слова – перевіряємо їх: знаходимо відсоток схожості пошукового запиту із кожним знайденим словом зі словника БД; знаходимо максимальний відсоток схожості; повертаємо всі слова з цим відсотком (на випадок, якщо кілька слів матимуть однаковий відсоток, який виявиться максимальним).

При кожному пошуку необхідно буде розраховувати відстань Левенштейна. Для цього потрібно знайти найшвидшу реалізацію алгоритму порівняння слів для MySQL.

Для MySQL є наступні реалізації цього алгоритму:

− запит в стилі алгоритму Левенштейна, автор – Gordon Lesti [6];

− функція для обчислення відстані Левенштейна, автор – Jason Rust [7];

− функція для обчислення відстані Дамерау-Левенштейна, написана на основі функції мови C, автор – Diego Torres [8].

Схожість двох рядків в MySQL визначається на підставі функції Торреса `damlevlim()`, тому що під час тестування вона показала найбільш швидкий результат.

Наприклад:

```
// знаходимо все рядки з різницею
// Дамерау-Левенштейна 0 або 1
$q = mysqli_query($conn, 'SELECT ' .
     'city_id, title_ru FROM _cities ' .
     'WHERE damlevlim("' . $input_m .
     '",metaphone,20)<2');
```

При цьому в PHP обчислення схожості кожного результату з пошуковим запитом здійснюється функцією `similar_text()`, яка повертає кількість співпадаючих символів в двох рядках. Така реалізація алгоритму не використовує стека, натомість застосовуються рекурсивні виклики, що в деяких випадках прискорює процес пошуку. Складність алгоритму становить $\sigma(N^3)$, де N – довжина найдовшого з двох рядків.

Під час експериментів було встановлено, що функція `similar_text()` повертає різні результати для слів на кирилиці й латиниці при однаковій відстані Левенштейна. Тому додатково необхідно використати функцію `utf8_to_extended_ascii()` (пошук усіх багатобайтових символів та їх перетворення у однобайтові), яка застосовується для вирішення тієї ж проблеми при використанні PHP-функції `levenshtein()` (лістінг 1).

Лістінг 1

```
// записуємо результати до масиву
while ($row = mysqli_fetch_assoc($q))
    $damlev_result[] = [ $row['city_id'], $row['title_ru'] ];
// якщо результатів більше 1, відбираємо з максимальною схожістю
if (count($damlev_result) > 1){
    foreach ($damlev_result as $v)
        // обчислюємо схожість кожного результату з пошуковим запитом,
        // результат записуємо в масив
        similar_text(utf8_to_extended_ascii($input,$charMap),
                      utf8_to_extended_ascii($v[1],$charMap), $similar_text_result[] );
    // обчислюємо максимальну схожість
    $max_similarity = max($similar_text_result);
    // обчислюємо ключі результатів з максимальною схожістю
    $most_similar_strings=array_flip(array_keys($similar_text_result, $max_similarity) );
    // повертаємо результати з цими ключами
    return array_intersect_key($damlev_result,$most_similar_strings);
}
// якщо результатів немає або він 1,
// повертаємо порожній масив або масив з 1 //результатом
else return $damlev_result;
```

Відстань Левенштейна або відстань Дамерау-Левенштейна – алгоритми, які забезпечують мінімальну кількість операцій для перетворення одного рядка в інший. Левенштейн запропонував операції вставки, видалення та заміни одного символу, а Дамерау доповнив їх операцією транспозиції, тобто коли два сусідніх символу міняються місцями.

Нехай S_1 та S_2 – два рядки, що мають довжину M та N відповідно над деякими алфавітом, тоді редакційну відстань (відстань Левенштейна) $d(S_1, S_2)$ можна розрахувати за наступною рекурентною формулою [9]:

$$d(S_1, S_2) = d(M, N), \text{ де}$$

$$D(i,j) = \begin{cases} 0; i = 0; j = 0 \\ i, j = 0, i > 0 \\ j, i = 0, j > 0 \\ \mu_{ij}; j > 0; i > 0 \end{cases},$$

де $\mu_{ij} = \min\begin{pmatrix} D(i, j-1)+1, \\ D(i-1, j)+1, \\ D(i-1, j-1)+m(S_1[i], S_2[j]) \end{pmatrix}$

і $m(a,b)$ дорівнює нулю, якщо $a = b$, інакше одиниці. $\min(a,b,c)$ повертає найменший із аргументів.

Для визначення послідовності операцій, необхідних для переходу від одного слова до іншого, потрібно знайти найкоро-

тший шлях від першої [0,0] клітинки матриці до останньої $[i, j]$. У PHP цей алгоритм реалізовано функцією `levenshtein()`.

Висновки і перспективи досліджень

Використовувати розрахунок відстані Левенштейна в MySQL необхідно тільки у випадках, коли рядок, з яким потрібно порівнювати, короткий, а таблиця зі словами, які підлягають порівнянню з рядком – невелика. В іншому випадку, – у разі таблиці з великим словником, можливим рішенням може бути її поділ на декілька таблиць, наприклад, за першою буквою, або за довжиною слова або його метафоном. Прискорення розглянутого алгоритму нечіткого пошуку в тексті або словнику можливе, якщо: впорядкувати за алфавітом базу даних; створити індекс-файл за типом глосарію (впорядковані за алфавітом унікальні слова з усього тексту файлу бази з посиланнями на потрібний рядок (-ки) в базі).

Наступним етапом є впровадження обчислювальної технології нечіткого пошуку в тексті в якості модуля пошукової системи веб-сайту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Фролов, А.С. Разработка алгоритма нечеткого поиска на основе хэширования [Електронний ресурс] / А.С. Фролов // Молодой ученый. – 2016. – №13. – С. 357-360. – Режим доступу: https://moluch.ru/archive/117/32158/ (дата звернення: 27.03.2018)
2. Желудков, А. В. Особенности алгоритмов нечёткого поиска [Текст] / А.В. Желудков, Д.В. Макаров, П.В. Фадеев // Инженерный вестник МГТУ им. Н.Э. Баумана. – Москва, 2014. – С. 502-503.
3. Харитоненков, А.В. Поиск на неточное соответствие: коды Хемминга [Електронний ресурс] / А.В. Харитоненков. – Режим доступу: http://www.jurnal.org/articles/2009/inf32.html (дата звернення: 27.03.2018)
4. Лиманова, Н.И. Алгоритм нечеткого поиска в базах данных и его практическая реализация [Текст] / Н.И. Лиманова, М.Н. Седов. // Сборник трудов III международной конференции и молодежной школы «Информационные технологии и нанотехнологии» (ИТНТ-2017). – Самара: Новая техника, 2017. – С. 1885-1889.
5. Міняйло, А.Ю. Використання відстані левенштейна для аналізу подібності даних [Електронний ресурс]. – Режим доступу: http://www.pm-mm.dp.ua/index.php/pmmm/article/download/111/111 (дата звернення: 27.03.2018)
6. Gordon Lesti. Fuzzy Fulltext Search with Mysql [Електронний ресурс]. – Режим доступу: https://gordonlesti.com/fuzzy-fulltext-search-with-mysql/ (дата звернення: 28.02.2018)
7. Levenshtein distance// from the Artful Common Queries page [Електронний ресурс]. – Режим доступу: http://www.artfulsoftware.com/infotree/qrytip.php?id=552 (дата звернення: 28.02.2018)
8. Damerau-Levenshtein Distance UDF for MySQL [Електронний ресурс]. – Режим доступу: https://github.com/ifsnop/damlev (дата звернення: 28.02.2018)
9. Відстань Левенштейна // Вікіпедія: вільна енциклопедія [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Відстань_Левенштейна (дата звернення: 28.02.2018)

Kateryna KUZMA
Mykolayiv

THE COMPUTING TECHNOLOGY OF FUZZY SEARCH IN A TEXT OR IN A DICTIONARY

The paper deals with the computing technology for fuzzy searching implemented for the MySQL database using PHP functions. The proposed technology is based on the calculation of the Damerau-Levenstein distance. The Levenshtein algorithm have been analyzed, the concept of metric of strings similarity was considered.

It has been determined that computing technology works effectively when comparing a search query with a small-sized dictionary. It is appropriate to use it to search for names according to an existing dictionary, in which it is highly probable that a user may make a mistake of at least one character. As a result of the research, it has been found that, since MySQL today does not offer an embedded solution for fuzzy searches in a text or in a dictionary, development of effective computing technologies capable for expanding the functionality of website search engines is actual.

***Keywords**: fuzzy string search, Damerau-Levenstein distance, computing technology, string similarity, MySQL.*

Екатерина КУЗЬМА
Николаев

ВЫЧИСЛИТЕЛЬНАЯ ТЕХНОЛОГИЯ НЕЧЕТКОГО ПОИСКА В ТЕКСТЕ ИЛИ СЛОВАРЕ

В работе рассмотрена вычислительная технология нечеткого поиска, реализованная для СУБД MySQL с использованием функций PHP. Предложенная технология базируется на вычислении расстояния Дамерау-Левенштейна. Выполнен анализ алгоритма Левенштейна, рассмотрено понятие метрики сходства строк.

Определено, что вычислительная технология эффективно работает при сравнении поискового запроса с небольшим по размеру словарем. Ее следует использовать для поиска названий и имен в имеющимся словаре, при котором высока вероятность того, что пользователь может совершить ошибку набора или ошибиться хотя бы на один символ. В результате проведенного исследования установлено, что поскольку на сегодня MySQL не предлагает встроенного решения для реализации нечеткого поиска в тексте или словаре, актуальными являются вопросы разработки эффективных вычислительных технологий, способных расширить функциональность поисковых систем веб-сайтов.

***Ключевые слова**: нечеткий поиск, расстояние Дамерау-Левенштейна, вычислительная технология, сходство строк, MySQL.*

Стаття надійшла до редколегії 28.03.2018