УДК 519.6

Веприцький Д. Ю., магістр спеціальності 126 «Інформаційні системи та технології»
Шевченко Н. Ю., к.е.н., доцент, доцент кафедри інтелектуальних систем прийняття рішень

РОЗРОБКА КОНЦЕПТУАЛЬНИХ ПІДХОДІВ ДО АНАЛІЗУ ТЕКСТОВИХ ДАНИХ

Донбаська державна машинобудівна академія, м. Краматорськ

В залежності від сфери діяльності аналіз текстової інформації буде різнитися. При цьому аналіз зазвичай повинен вирішувати наступні задачі: визначати стилістику, статистичні характеристики тексту (довжину, наявність та кількість повторів, та ін.), а також оцінювати унікальність текстової інформації. Аналіз текстової інформації в загальному виді умовно можна розділити на кілька етапів.

Етап 1. Початковий аналіз текстових даних, який передбачає отримання інформації та проведення основних обробок. На даному етапі виконується: визначення стилю тексту; виявлення частин мови в тексті. Для визначення стилю за допомогою програмного забезпечення необхідно визначити загальну кількість слів у заданому тексті.

Псевдокод алгоритму визначення кількості слів:

```
    <? php
for ( $i = 192; $i < 256; $i++ ) {
  $abc. = chr($i); //Цикл на каждом шагу добавляет к переменной $abc новую букву
}
$abc=iconv( 'cp1251', 'utf-8', $abc); //Переводим строку из кодировки utf-8 в cэ1251
echo 'Количество слов в тексте: ', str_word_count($text,0,$abc);?>
```

Кількість частин мови можна визначати за допомогою спеціальної функції chastrechiRUS () з пошуку відповідних закінчень слів.

Етап 2. SEO-аналіз текстових даних передбачає перевірку якості та релевантності тексту за словами та словосполученнями (колокаціям), що містяться у ньому. На даному етапі визначається довжина тексту та відбувається пошук посилань та ключових слів.

Псевдокод алгоритму вимірювання довжини тексту:

```
<?php
        $text=$_POST['text'];
        $text_nonspace=str_replace(array(" "), '', $text);
        echo 'Количесвто символов с пробелом: ' , mb_strlen($text, 'utf-8');
        echo '<br/>','Количество символов без пробела: ' , mb_strlen($text_nonspace, 'utf-8');
        ?>
```

Для пошуку ключових слів використовується функція get_tag (), у якій описується частина коду що дозволяє знаходити задані html теги.

Частота появи ключових слів визначається за формулою:

$$X = \frac{M}{N} \cdot 100\%, \quad (1)$$

де X – частота; M – кількість повторів слова або ключової фрази; N – загальна кількість слів.

Ключові слова можуть не обертатися в спеціальні html теги, а визначатися як найчастіше повторювані слова і словосполучення в тексті.

Для цього визначається відсоток повторів як відношення кількості речень (фраз) з повторами до загальної кількості речень (фраз) в тексті:

$$D_{повтор}^{k} = \frac{\sum_{j=1}^{n} P_j^{повтор}}{\sum_{i=1}^{m} P_i} \cdot 100\%, \; P_j^{повтор} = \begin{cases} 0, якщо\ в\ реченні\ відсутні\ повтори; \\ 1, якщо\ в\ реченні\ є повтор. \end{cases}$$

де $D_{повтор}^{k}$ – відсоток повторів у k-м документі; $\sum_{j=1}^{n} P_j^{повтор}$ – кількість речень (фраз) з повторами; j – індекс речення з повторами; n – кількість речень з повторами; $\sum_{i=1}^{m} P_i$ – кількість речень (фраз) у документі;

Для пошуку посилань використовується регулярне висловлювання: preg_match_all('/(<a[^>]*)href=(\»?)([^\s\»>]+?)(\»?)([^>]*>)/ismU',$text,$res).

Етап 3. Перевірка унікальності одного тексту по відношенню до іншого [1] за допомогою алгоритму шинглів (рис. 1).

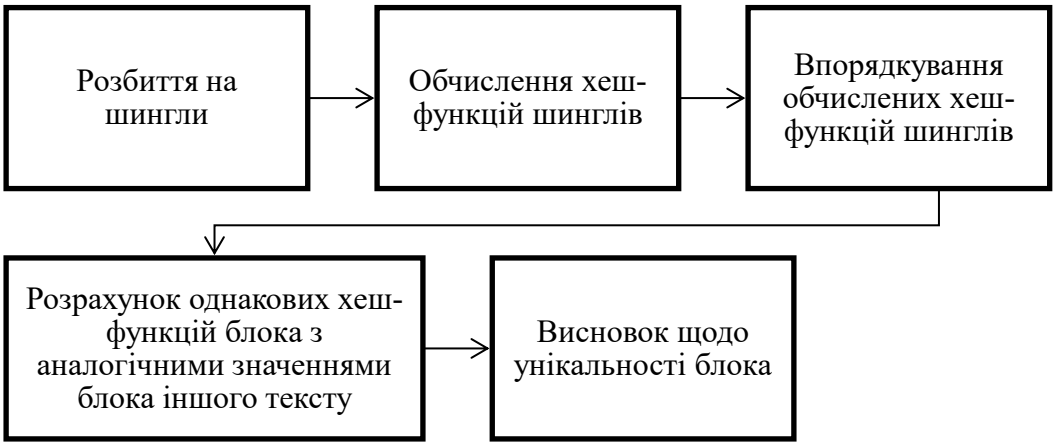

Рисунок 1 – Етапи роботи алгоритму

Для визначення хеш-функцій пропонується використати алгоритм *CRC*.

Даний алгоритм базується на властивостях ділення з залишком двоїстих багаточленів, тобто багаточленів над кінцевим полем. Значення CRC є залишком від ділення багаточлена, що відповідає вхідним даним, на деякий фіксований породжуючий багаточлен:

$$\sum_{n=0}^{N-1} a_n \cdot x^n. \quad (2)$$

Кожній кінцевій послідовності бітів $a_0, a_1, ...a_{N-1}$. взаємно однозначно зіставляється двоїстий поліном, послідовність коефіцієнтів якого представляє собою вихідну послідовність.

Кількість багаточленів степені менше *N* дорівнює 2^N, що співпадає з числом всіх двоїстих послідовностей довжини *N*.

Значення контрольної суми з породжуючим багаточленом *G(x)* степені *N* визначається як бітова послідовність довжини *N*, яка представляє багаточлен *R(x),* який отриманий в залишку при діленні багаточлена *P(x)*, який є вхідним потоком біт, на багаточлен *G(x):*

$$R(x) = P(x) \cdot x^N \bmod G(x), \quad (3)$$

де *R(x)* – багаточлен, що представляє значення *CRC*; *P(x)* – багаточлен, коефіцієнти якого представляють вхідні дані; *G(x)* – породжуючий багаточлен; *N* – степінь породжуючого багаточлену.

Відсоток унікальності тексту по відношенню до іншого тексту буде визначатися за формулою:

$$P_{УНІК} = \frac{\sum n - \sum n_i}{\sum n} \cdot 100\%, \quad (4)$$

де $P_{УНІК}$ – відсоток унікальності тексту по відношенню до іншого тексту; $\sum n$ – сума усіх шинглів тексту, який перевіряється; $\sum n_i$ – сума однакових шинглів в тексті 1 та в тексті 2.

Для визначення того, чи є два речення парафразами, тобто однаковий чи їх сенс для носіїв мови,введемо числову міру подібності – коефіцієнт Жаккара (K).

Якщо два речення A і B містять відповідно n(A) і n(B) лексичних одиниць, то:

$$K = c/(a + b - c), \quad (5)$$

де *a* – кількість символів в першому рядку; *b* – кількість символів у другому рядку; *c* – кількість співпадаючих символів.

У відповідності з цим критерієм міра подібності визначається як відношення кількості збіжних одиниць до загального числа різних одиниць.

Коефіцієнт Жаккара може знаходитися у діапазоні від 0 до 1 або від 1% до 100%.

Отже, наведені вище алгоритми аналізу текстової інформації передбачають, що якісна оцінка текстів повинна включати початковий аналіз текстових даних, seo-аналіз і оцінку його унікальності по відношенню до іншого тексту.

Список літератури

1. U. Manber. Finding Similar Files in a Large File System / U. Manber. – Winter USENIX Technical Conference, 1994. – 105 p.