УДК 656.7.085:657.71(045)

ЛІНГВІСТИЧНИЙ АНАЛІЗАТОР ТЕКСТІВ НА ЗАПОЗИЧЕННЯ З ВИКОРИСТАННЯМ СЛОВНИКІВ

О. Є. Литвиненко, А. А. Андрущенко

Національний авіаційний університет

litvinen@nau.edu.ua

Наведено принцип та проаналізовано ефективність роботи лінгвістичного аналізатора текстів на запозичення. Обґрунтовано важливість використання словників у програмах даного роду. Запропоновано методи для роботи зі словниками. Подано переваги і недоліки використання словників.

The article is dedicated to the development of software text-analyzer which uses dictionaries in its operation process. Advantages and disadvantages of the use of dictionaries are analyzed, several methods of data processing algorithms are offered.

Вступ

Новаторська розробка кафедри комп'ютеризованих систем управління Інституту комп'ютерних технологій — лінгвістичний аналізатор «Антиплагіат» зарекомендували себе унікальною системою перевірки текстів на запозичення. Є декілька шляхів удосконалення роботи цієї системи. Одним із них є використання спеціальних словників для роботи програми. Інтеграція словників до роботи системи дасть змог розпізнавати запозичення з використанням синонімічних рядів та іншомовних аналогів слів, а також допоможе збільшити ефективність самої системи. Під словником мається на увазі структура даних що містить інформацію про слова, їх синоніми, іншо-мовні аналоги та надає можливість кодувати змісти текстів. Використання схожих словників може також бути використано під час розробки алгоритмів пошукових систем. Спираючись на перелічені вище підстави, в цьому дослідженні буде створено власну модель аналізатора текстів на запозичення із використанням словників.

Мета

Метою дослідження є:

- розробка лінгвістичного аналізатора, що ґрунтується на використанні словників під час своїй роботи;
- розробка алгоритму порівняння текстів на запозичення шляхом пошуку спільних послідовностей слів у текстах.
- дослідження переваг використання словників в аналізаторі текстів на запозичення;
- розробка методів використання словників у роботі аналізатора;
- аналіз роботи аналізатора, ефективність використання ресурсів;
- аналіз використання системних ресурсів (обсяг оперативної пам'яті для функціонування аналізатора та постійної пам'яті для збереження словників, використання процесорного часу на виконання роботи з текстами) аналізатором при роботі зі словниками;
- аналіз подальших шляхів поліпшення роботи аналізатора.

Дослідження

У ході роботи було розроблено спрощену систему лінгвістичного аналізатора текстів на запозичення. Під час розробки було використано засоби сучасних мов програмування, а саме — C# .NET [2]. Розроблена система ґрунтується на використанні словників. Структуру словника показано на рис. 1.

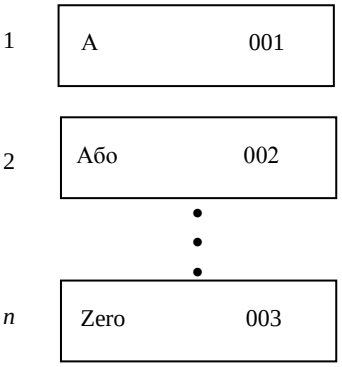

Рис. 1. Структура словника

Основним структурним елементом словника є словарна картка, що містить слово та його код. Під час обробки текстів аналізатором слова заміняються на відповідні коди. Синоніми та іншомовні аналоги слів мають однакові коди, тобто аналізатор сприйматиме їх як рівноцінні за змістом слова.

Для більш об'єктивного оцінювання плагіату в текстах можна використовувати різні словники для перевірки. Таким чином, функція підстановки синонімів може бути вимкнена задля порівняння отриманих результатів. Далі наведено алгоритм роботи аналізатора:

1. Тексти для аналізу завантажуються в пам'ять програми.
2. Кожен текст розбивається на слова, відбувається оновлення словника аналізатора.
3. Слова в текстах заміняються на відповідні коди статей словника.

4. Отримані послідовності кодів обробляються алгоритмом перевірки на запозичення, результатом якого є матриця перетинів текстів.

5. З матриці видаляються всі коротші, ніж п'ять слів, однакові послідовності.

6. Створюється єдина маска запозичень, що надає інформацію про співвідношення запозиченої інформації у тексті до повного обсягу інформації в тексті.

Вхідним форматом програми є текстовий файл «*.txt», збережений у кодуванні *Unicode*.

Існує декілька підходів до використання словників:

- використовувати один чи декілька базових словників і доповнювати їх зміст під час аналізу наступних текстів. Такий підхід надає можливість утворювати бази даних текстів для порівняння, в якій тексти зберігатимуться вже в закодованому вигляді. Це допоможе досягти кращої швидкодії при перехресній перевірці на плагіат великої кількості текстів та раціоналізувати процес зберігання вже перевірених текстів. Цей підхід має і свої недоліки: обсяг словників невпинно зростатиме, разом з цим і зростатиме обсяг пам'яті, необхідний для зберігання закодованих текстів. Зростатиме і час для кодування текстів.
- натомість можна утворювати новий словник для кожної пари текстів, що перевіряються. Отже, можна виграти в затратах системних ресурсів (пам'ять, процесорний час), але унеможливлюється повторне використання вже закодованих текстів.

Алгоритм перевірки текстів працює таким чином: два тексти пропускаються один через одного за допомогою зсуву одного тексту через інший. Зсув відбувається на одне слово за ітерацію. У кожній ітерації відбувається аналіз слів на їх перетині. Таким чином заповнюється матриця перетинів, де кількість рядків — це кількість зсувів (кількість слів у другому тексті), а кількість стовпчиків — кількість слів у першому тексті.

Алгоритм знаходить усі входження слів одного тексту в другий, але нас цікавлять змістові запозичення, в даному разі — послідовності слів. Було обрано послідовність з п'яти слів за основу. Оскільки саме така кількість зв'язаних слів може відобразити унікальний змістовий ланцюг [1]. Кінцева маска запозичень містить інформацію саме про використання однакових послідовностей слів у текстах, що порівнюються. Термін «маска» вжито в сенсі подібності певної структури до бітової маски та можливості схожого подальшого використання.

Наповнення словників. З кожним текстом, що проходить через аналізатор, відбувається автоматичне наповнення словника новою лексикою, якщо така наявна у конкретному тексті. Додавання синонімів та іншомовних аналогів можливе через спеціальний програмний інтерфейс.

Аналіз використання ресурсів та швидкодії роботи розробленої програми. Для аналізу роботи програми було використано курсові та лабораторні роботи, а також тексти іншої тематики. Загальний обсяг одинадцяти опрацьованих текстів — 17 131 слово. Було утворено новий словник обсяг якого становив 3178 слів. Графік наповнення словника показано на рис. 2.

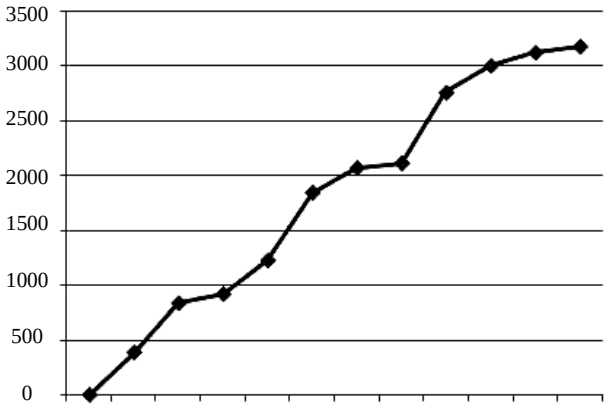

Рис. 2. Графік зростання обсягу словника

Видно, що збільшення обсягу словника зменшується з кожним новим текстом.

Обсяг новоутвореного словника — 38 кілобайт.

Тепер проаналізуємо швидкодію роботи лінгвістичного аналізатора.

Тест 1. Перевіримо два тексти: 2758 і 1913 слів відповідно. Словник порожній. Час від запуску програми до завершення аналізу — 4,859 с.

Час, витрачений на аналіз — 0,273 с.

Той самий тест виконаний з використанням великого словника:

Час від запуску програми до завершення аналізу — 4,984 с. Час витрачений на аналіз — 0,272 с.

Видно, що найбільш повільна частина роботи програми — це процес кодування текстів та роботи зі словником (пошук по словнику, додавання статей у словник).

Тест 2. Запустимо процес перевірки тих самих текстів на запозичення декілька разів.

Час, витрачений на одне коло аналізу — 0,273 с. Час, витрачений на два кола аналізу — 0,468 с. Час, витрачений на три кола аналізу — 0,741 с.

Тест 3. Перевіримо залежність швидкодії роботи аналізатора від обсягу текстів, що перевіряються. Для зручності тексти будемо перевіряти сам-на-себе.

Результати зручно представити у вигляді графіка (рис. 3). По осі абсцис відкладаємо розміри текстів (кількість слів), а по осі ординат — час, витрачений аналізатором за секунду.

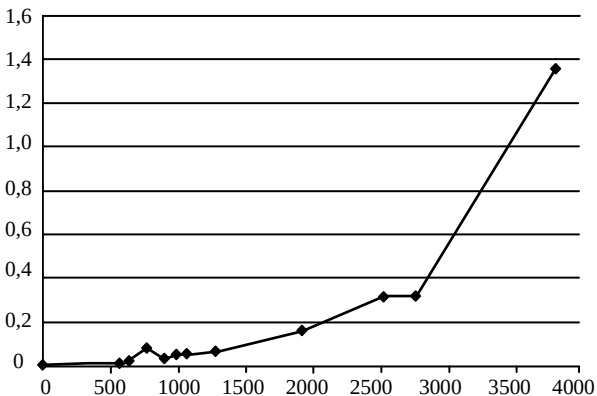

Рис. 3. Графік швидкодії аналізатора

Декілька точок зникають з графіка. Це можна пояснити роботою інших процесів у середовищі операційної системи. Загалом відслідковується квадратична залежність між кількістю слів у тексті та часом, що алгоритм витрачає на його перевірку.

Висновки

Під час дослідження було створено лінгвістичний аналізатор, що спирається під час своєї роботі на використання словників. Використання словників дає змогу аналізатору розпізнавати запозичення, що містять синонімічні ряди та заміну слів на їх іншомовні аналоги. Також використання словників дає нові можливості для економічного зберігання текстів у базі даних лінгвістичного аналізатора. Експериментальне дослідження роботи аналізатора виявило, що найслабкішим місцем з погляду швидкодії є процес кодування тексту та розширення лексичного обсягу словника. У цьому напрямі необхідно провести роботу щодо оптимізації. Алгоритм аналізатора текстів також може бути оптимізовано за рахунок використання більш швидких компіляторів або мов програмування низького рівня.

Загалом, алгоритм показав задовільні результати враховуючи, що процес тестування проводився на комп'ютері з середніми показниками швидкодії (Intel Celeron 2GHz, 2GHz 2GB RAM). Іншим напрямом роботи може бути процес інтеграції системи зі GPL/GNU словниками *ispell*.

ЛІТЕРАТУРА

1. *Анисимов А. В.* Компьютерная лингвистика для всех: Мифы. Алгоритмы. Язык / А. В. Анисимов. — К. : Наук. думка, 1991. — 208 с.

2. *Рихтер Дж.* CLR via C#. Программирование на платформе Microsoft .NET Framework 2.0 на языке C#. Мастер-Класс. Пер. с англ./ Дж. Рихтер — М.: Издательство «Русская Редакция»; С.Пб. : Питер, 2007. — 656 с.

Стаття надійшла до редакції 28.09.09.