УДК 004.021:81

Є. Левус[1], С. Бук[2], Є. Яворський[1]
[1]Національний університет "Львівська політехніка",
кафедра програмного забезпечення
[2]Львівський національний університет імені Івана Франка,
кафедра загального мовознавства

АЛГОРИТМ ВІДОБРАЖЕННЯ ЗМІНИ ЛЕКСИЧНОЇ НАСИЧЕНОСТІ ТЕКСТУ

Описано запропонований алгоритм виявлення зміни відношення кількості різних слів до загальної кількості слів у тексті, який можна використовувати для вирішення питань визначення авторського стилю. Проблема порівняння стилів текстових творів є актуальною в наукових дослідженнях, зокрема в інформатиці. За допомогою цих методів можна покращити якість класифікації та впорядкування текстових колекцій, що актуально для пошукових систем і великих сховищ текстових даних. Відмінною ознакою алгоритму від аналогічних є його можливість аналізувати динаміку лексичної насиченості по тексту. Алгоритм програмно реалізовано в системі аналізу текстів.

Ключові слова: лексична насиченість, алгоритм, авторський стиль, слово.

Described is the algorithm of detecting changes in the ratio of different words to the total number of words in the text which can be used to address the issues of determining the author's style. The problem of comparing text styles works is relevant in both philological and historical studies, as well as in computer science. The use of these comparison methods can improve the quality of classification and text collections management, which is important for search engines and large repositories of text data.

A distinctive feature of the algorithm among similar ones is its ability to analyze the dynamics of lexical richness of the text. The algorithm is implemented in software system for texts analysis.

Key words: algorithm, lexical richness, author's style, word.

Питання аналізу тексту стосовно авторського стилю

Сьогодні чималу увагу в галузі прикладної лінгвістики приділяють питанню аналізу тексту з погляду особливостей використаного в ньому авторського стилю. Проблема порівняння стилів текстових творів є актуальною в багатьох сферах людської діяльності. В історичних дослідженнях аналіз стилю застосовують для визначення авторства або часу створення історичних документів, у філологічних дисциплінах – для вивчення стилістичних особливостей текстів чи мови творів різних жанрів, авторів тощо. На практиці завдання порівняння стилів текстів актуальні у криміналістиці, наприклад, для встановлення автора письмової погрози або визначення його індивідуальних особливостей під час проведення оперативно-пошукових заходів.

Кількісні підходи до вирішення таких проблем нині особливо актуальні, оскільки вони дозволяють автоматизувати процедуру порівняння стилів текстів, дати формалізоване об'єктивне рішення. Розвиток цих підходів важливий також і для інформатики, оскільки за їх допомогою можна підвищити якість класифікації та впорядкування текстових колекцій, що актуально для пошукових систем і великих сховищ текстових даних. Порівняння стилів текстів проводиться, зазвичай, на основі сукупності ознак, що відображають властивості стилів текстів. Зазвичай розглядаються частотні ознаки (частоти появи певних слів, буквосполучень тощо), які можуть бути легко формалізовані для проведення за їх допомогою кількісного (частотного) аналізу текстів [1].

Незважаючи на велику кількість підходів до розв'язання задачі встановлення авторського стилю, існують мало або й зовсім не досліджені методи отримання його числових характеристик. Зокрема, немає алгоритмів, які б могли обчислити зміну лексичної насиченості. Лексична насиченість (словникова різноманітність, лексична щільність[2]) – відношення кількості різних слів до загальної кількості слів у тексті, використовується як одна величина, що характеризує текст (вводиться терміносполука «лексична насиченість», бо наявні іменування дещо відрізняються за суттю – в них строго використовується загальна кількість слів у тексті [2], а в цій роботі використана кількість слів у обраному текстовому блоці). Тому доцільно розробити такий алгоритм з метою використання результатів обчислення як додаткових параметрів авторського стилю. Графічне зображення зміни насиченості у тексті великого обсягу дозволить зробити висновок про особливості авторського стилю.

Існуючі методи вирішення

У [3] значення лексичної насиченості текстів використовується як допоміжне число під час психолінгвістичного аналізу текстів, і фігурує лише як одне число на один текст. Також це значення використовується як допоміжна величина під час пошуку інформації у [2].

Доступні два функціонально-завершені некомерційні програмні засоби, що реалізують задачу знаходження лексичної насиченості тексту. Обидва продукти мають веб-інтерфейс.

Wordcounter [4] опрацьовує тексти лише англійською мовою, характеризується можливостями виведення списку найчастіше вживаних слів та налаштовуваністю базових функцій. Проте істотним недоліком є те, що порівняння слів відбувається не за інфінітивом, а за основою, котра обчислюється просто шляхом відкидання останніх декількох літер. Тому слова, в яких відбувається заміна букв під час формотворення, не збігатимуться.

Document Information Tool [5] також опрацьовує тексти лише англійською мовою, здійснює вивід статистики слів та літер. Але при аналізі взагалі не беруться до уваги слова з однаковою інфінітивною формою, вони вважаються різними.

Постановка задачі

Необхідно розробити та реалізувати алгоритм відображення зміни лексичної насиченості тексту. Зокрема під час реалізації потрібно впровадити підтримку словників різних мов для універсальності програмного засобу.

Рішення

Серед наявних у вільному доступі словників із словоформами, було обрано словники формату Hunspell [6], оскільки в цьому форматі можна знайти словники для більшості сучасних мов, серед яких і українська. Очевидно, що для обчислення насиченості тексту, необхідно перетворити усі слова тексту в інфінітивні форми, а тоді обчислити частку від ділення кількості різних слів в утвореному масиві на загальну довжину масиву.

Загальний формат даних у словниках Hunspell такий:

слово/abc,

де abc – перелік класів, згідно з якими із слова можна утворити словоформи (правила перетворення містяться у додатковому афіксному файлі).

Для підвищення продуктивності перетворення тексту в набір словникових форм, під час операції ініціалізації алгоритму необхідно сформувати всі можливі форми для кожного слова.

Отже, алгоритм знаходження насиченості для текстового блока виглядає так:

1) ініціалізація словника;
2) поділ вхідного тексту на слова;
3) перетворення кожного слова вхідного тексту на його словникову форму; слова, що не відомі програмі, відкидаються (такий підхід мінімально спотворює вплив невідомих слів на результат аналізу насиченості);
4) обчислення Np. – кількості різних слів у результуючому наборі словникових форм;
5) обчислення насиченості – Np. / Nзаг.

Отримана функція обчислення насиченості використовується для виявлення зміни насиченості протягом написання тексту автором.

Розв'язання такої задачі ґрунтується на виділенні окремих груп певної кількості слів (рис. 1).

Рис. 1. Обчислення зміни лексичної насиченості

Тут $\mathbf{a_i}$ – слово із тексту; $\mathbf{K_j}$ – обчислене значення насиченості для j-ї групи слів довжиною **n,** $j = \overline{1, N}$. Фактично, відбувається просування блока, що відповідає за початок та кінець тексту для обчислення насиченості, на одне слово для кожної нової точки графіка (масиву значень). Тому кількість точок на результуючому графіку становитиме $D = N - n + 1$.

Алгоритм обчислення зміни насиченості:

1) отримання блока перших n слів із перетворених у словникові форми слів тексту;
2) обчислення насиченості одержаного блока та збереження її в результуючий масив;
3) якщо останнє слово блока – це останнє слово перетвореного тексту, то завершити процедуру;
4) відкинути перше слово та перейти на пункт 1);
5) для зручності інтерпретації результатів аналізу тексту фахівцями варто відображати графік залежності лексичної насиченості блока від позиції у тексті.

Проаналізувавши декілька творів при різних значеннях n, було визначено, що аналіз графічного результату варто проводити, коли n=500. При значно більших чи значно менших значеннях графік згладжується, адже різниця в насиченості стає мінімальною (у разі великих значень n вона завжди низька, а малих значень n – завжди висока).

Запропонований алгоритм був реалізований у програмному продукті MorphAnalyzer (рис. 2–5) засобами середовища Eclipse IDE із плагіном WindowBuilder [7]. Віконний інтерфейс будувався із використанням JFace+SWT [8]. Застосовувались принципи об'єктно-орієнтованого програмування та модульного програмування.

Програмний продукт MorphAnalyzer дозволяє проаналізувати вхідний текст та вивести деяку кількісну, частотну тощо інформацію про нього. Після запуску програми необхідно вибрати файли словників та натиснути кнопку «Ініціалізація» для того, щоб словники завантажились в пам'ять.

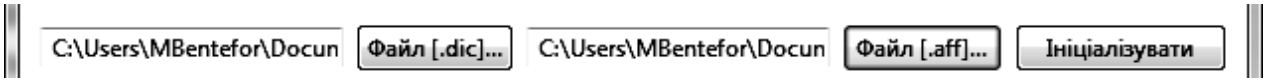


Рис. 2. Вибір файлів словника та ініціалізація

Залежно від робочої системи, ця процедура може тривати до 30 с. Після завершення, програма виведе інформацію про отримані словникові дані.

Рис. 3. Результат ініціалізації

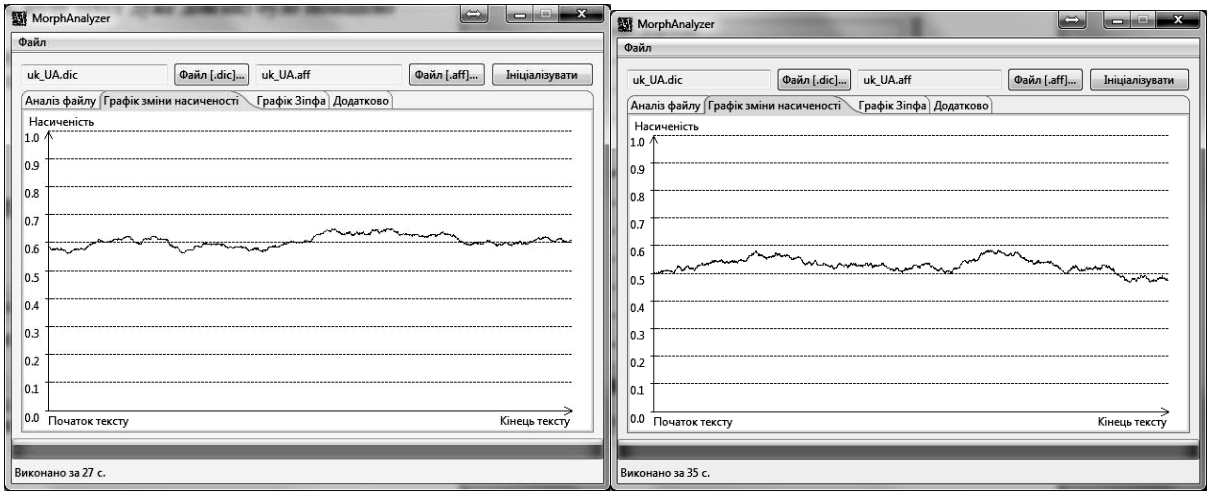
Після цього потрібно вибрати вхідний текстовий файл і, натиснувши кнопку «Аналізувати», опрацювати тексти. Часові витрати на виконання операції аналізу залежать від розміру вхідного файла. На особливо великих обсягах інформації процес аналізу може зайняти до години часу (таблиця).

Результати аналізу текстів

Розмір тексту (слів)	Час виконання аналізу
1129 (науковий допис)	36 секунд
3534 (стаття)	77 секунд
21650 (повість)	5 хвилин 31 секунда
225311 (роман)	43 хвилини

Тестували на робочій машині із процесором Intel Core i7 (4 ядра, 2.4 ГГц) під управлінням ОС Windows 7.

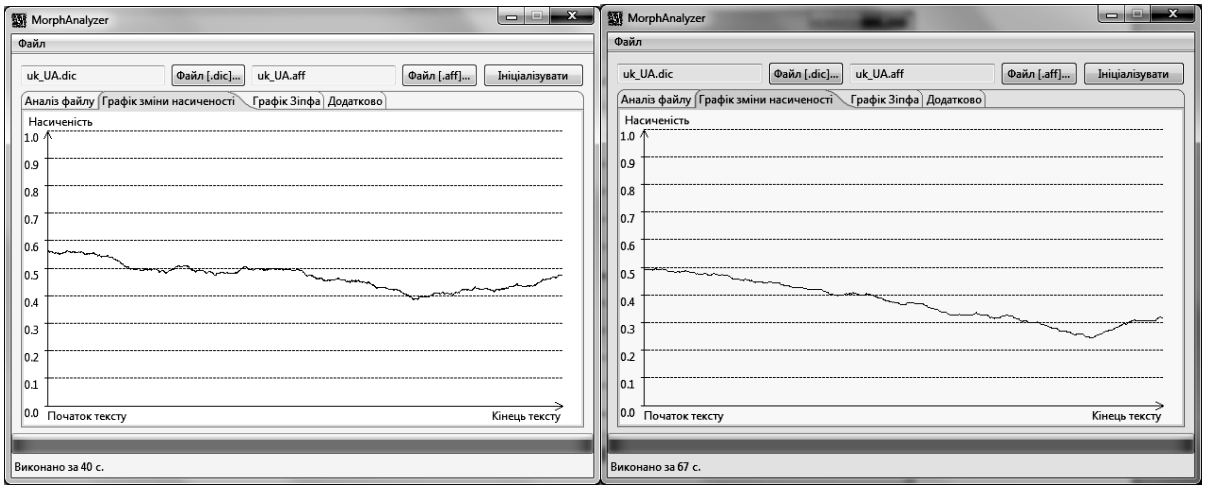
Як видно з проведених обчислювальних експериментів, стиль текстів визначає характер результуючого графіка. Результати аналізу художніх творів наведено на рис. 4.



а *б*

Рис. 4. Результат аналізу твору Т.Г. Шевченка «У всякого своя доля» (а); результат аналізу твору «Велесова книга» (б)

Наявність великої кількості повторень у текстах наукового характеру проявляється у загальному спуску графіка (рис. 5).



а *б*

Рис. 5. Результат аналізу наукової статті для галузі ІТ (а); результат аналізу наукової статті для галузі радіоелектроніки (б)

Отже, програмне застосування алгоритму можна використовувати для подальших лінгвістичних та психолінгвістичних досліджень.

Висновки

У роботі розглянуто актуальність питання визначення авторського стилю в різних галузях наукових досліджень. Розроблено алгоритм, що дає змогу отримати допоміжні параметри для виконання таких завдань.

Алгоритм реалізовано у програмному застосунку, основною функцією якого є візуалізація графіка зміни лексичної насиченості. За допомогою розробленого програмного забезпечення можна знаходити спільні ознаки в творах одного автора, отримуючи нову характеристику авторського стилю. Так, після аналізу невеликої кількості (близько десяти) прозових творів Івана Франка, було зроблено висновок, що на відміну від інших авторів, насиченість у творах письменника коливається у доволі широкому діапазоні.

Отримані результати аналізу текстів демонструють перспективність використання розробленого алгоритму у багатьох прикладних галузях, де використовуються психолінгвістичний та лінгвістичний аналіз текстів [9–12]. Надалі планується розвинути наявну програмну реалізацію з розробкою та впровадженням нових алгоритмів для автоматизації порівняння авторський стилів, беручи за основу графік зміни насиченості.

1. Шевелев О.Г. Разработка и исследование алгоритмов сравнения стилей текстовых произведений: Автореф. дисс. – Томск, 2006. 2. Верес М.М., Лемківський Є.О., Омельченко О.А. / Масово розподілений пошуковий робот // Проблеми інформаційних технологій. – 2011. – №1 (009). 3. Психолінгвістичний текстовий аналіз – Матеріал з Вікіпедії – вільної енциклопедії [Електронний ресурс]. – Режим доступу: http://uk.wikipedia.org/wiki/Психолінгвістичний_текстовий_аналіз (2013). 4. Wordcounter [Electronic resource]. – Web page: http://www.wordcounter.com/ (2004). 5. Character And Word Counter With Frequency Statistics Calculator [Electronic resource]. – Web page: http://www.csgnetwork.com/documentanalystcalc.html (2013). 6. Man hunspell – format of Hunspell dictionaries and affix files [Electronic resource]. – Web page: http://pwet.fr/man/linux/fichiers_speciaux/hunspell (2013).7. Eclipse Workbench User Guide [Electronic resource]. – Web page: http://help.eclipse.org/juno/index.jsp?nav=%2F0 (2012). 8. JFace Eclipse toolkit [Electronic resource]. – Web page: http://wiki.eclipse.org/index.php/JFace (2013). 9. Кыштымова И.М. / Психосемиотический анализ текста: диагностическое значение категории "время". – Режим доступу: http://www.lib.tsu.ru/mminfo/000085170/26/image/26-050.pdf. 10. Психология – Матеріал з Вікіпедії – вільної енциклопедії [Електронний ресурс]. – Режим доступу: http://ru.wikipedia.org/wiki/Психология (2013). 11. Горелов И. Н., Седов К. Ф. // Основы психолинг-вистики. — Москва. – 1997. 12. Засєкіна Л.В. Вступ до психолінгвістики / Л.В. Засєкіна, С.В. Засєкін. – Острог: Вид-во Нац. ун-ту «Острозька академія», 2002. – 168 с.