DOI: 10.20535/kpisn.2021.3.240780
УДК 004.021:004.91

Я.О. Юсин*, Т.М. Заболотня
КПІ ім. Ігоря Сікорського, Київ, Україна
*corresponding author: yusin.yakiv@gmail.com

ГЕНЕРУВАННЯ КОРПУСІВ ТЕКСТОВИХ ДАНИХ НА ОСНОВІ ДЕТЕРМІНОВАНОГО МЕТОДУ

Received: 27 Sep. 2021. Accepted: 7 Dec. 2021.

Проблематика. Розв'язання великої кількості задач у галузі оброблення природної мови передбачає використання корпусів текстових даних, що зумовлює актуальність підготовки останніх. Їх формування на основі справжніх текстів потребує часових витрат, а також є не завжди доцільним. Тому популярності набуває автоматизоване генерування корпусів на основі різних методів й алгоритмів, що значно спрощує підготовку експериментальних даних.
Мета дослідження. Збільшити кількість знаходжуваних дефектів під час тестування програмних реалізацій методів оброблення природномовних текстових даних, розробивши новий метод генерування корпусів текстових даних задля їх використання як вхідних даних для тестування.
Методика реалізації. Запропоновано детермінований метод генерації корпусів текстових даних CorDeGen (від Corpora Deterministic Generation), що задовольняє такі вимоги: детермінованість; залежність на вході лише від бажаної кількості термів у корпусі, що генерується; забезпечення генерування корпусу нетривіальної структури. На основі запропонованого методу розроблено відповідний алгоритм, а також виконано програмну реалізацію на платформі .NET (мова програмування – C#). За допомогою цієї програмної реалізації оцінено швидкодію та ефективність розробленого методу.
Результати дослідження. Оцінювання швидкодії розробленого методу CorDeGen показало степеневу залежність часу генерування корпусу від кількості термів (вхідного параметра) з показником степеня близьким до 1,5. У межах дослідження доцільність використання розробленого методу для тестування коректності програмних реалізацій показана на прикладі тестування методу кластеризації kсередніх.
Висновки. Тестування створеного детермінованого методу генерації корпусів текстових даних довело його ефективність при тестуванні інших методів оброблення природномовних даних, як-от кластеризації, замість природних корпусів.
Ключові слова: корпус текстових даних; генерація корпусів; оброблення природномовних даних; кластеризація даних; метод k-means; метод k-середніх.

Вступ

Велика кількість задач у галузі оброблення природної мови (natural language processing) пов'язана з перетворенням або аналізом текстових даних, оформлених як корпуси. У лінгвістиці корпусом (corpora) називають певним чином оброблену й анотовану колекцію текстів [1]. Корпуси можна класифікувати за певними ознаками: мовою, обсягом, одномовністю чи багатомовністю текстів тощо [2]. У межах цього дослідження розглянуто природні та генеровані корпуси: природні складаються з текстів, створених людиною (літературні твори, дописи в соціальних мережах тощо), а генеровані складаються з текстів, отриманих частково чи повністю автоматично.

До задач, безпосередньо пов'язаних з аналізом корпусів, належать задачі кластеризації та класифікації, коли тексти заданого корпусу розподіляють за кластерами (чи класами) на основі певних ознак. Водночас результати застосування методів кластеризації чи класифікації на певних еталонних корпусах є відправною точкою для оцінювання їх ефективності. Це зумовлює необхідність підготовки якісних еталонних корпусів, бо інакше існує ризик отримати некоректні оцінки. Наприклад, якщо обраний

Рекомендуємо цитувати цю статтю так: Я.О. Юсин, Т.М. Заболотня, "Генерування корпусів текстових даних на основі детермінованого методу", Наукові вісті КПІ, № 3, с. 38–45, 2021. doi: 10.20535/kpisn.2021.3.240780.

Please cite this article as: Ya.O. Yusyn and T.M. Zabolotnia, "Text data corpora generation on the basis of the deterministic method", *KPI Science News*, no. 3, pp. 38–45, 2021. doi: 10.20535/kpisn.2021.3.240780.

еталонний корпус містить статистичні відхилення (нетиповий розподіл текстів і їх зв'язків, спотворену частотність слів тощо), то це може спотворювати (завищувати чи занижувати) отримувані оцінки.

Подібна ситуація може виникати при тестуванні програмних реалізацій методів оброблення природномовних текстових даних. Традиційні методології тестування потребують порівняння отриманого результату з очікуваним, що часто складно забезпечити, бо очікуваний результат потрібно фактично обчислити вручну; використовуючи ж генеровані корпуси, необхідно бути певним, що генерована структура корпусу охоплює всі можливі випадки.

Відтворюваність результатів є ще однією актуальною проблемою, пов'язаною з використанням природних або генерованих корпусів у задачах тестування й оцінювання методів оброблення природномовних текстових даних. Щоб результати можна було отримати повторно через певний проміжок часу або ж щоб їх могли відтворити інші науковці, потрібно: у випадку природних корпусів – розмістити їх у загальнодоступному місці та підтримувати їх доступність; у випадку генерованих корпусів – описати метод генерації (чи послатись на наявний) із поясненням усіх вхідних даних для генерації (і знову ж, якщо це певні природні тексти, – розмістити їх у загальнодоступному місці). Це може викликати певні ускладнення у разі великого обсягу вхідних даних, оскільки не кожен загальнодоступний майданчик готовий їх розміщувати, а власний необхідно підтримувати в доступному стані. Тож ці ускладнення можуть призводити до втрати вхідних даних певних досліджень, а отже й до втрати їх відтворюваності.

Задача створення нових корпусів текстових даних (або методів їх генерування) є актуальною та має практичну цінність, бо наслідками її розв'язання буде:

– підвищення якості тестування програмних реалізацій нових і наявних методів оброблення природномовних текстових даних за критерієм кількості знайдених дефектів (завдяки кращому охопленню тестових випадків і спрощенню визначення очікуваного результату);

– підвищення відтворюваності результатів тестування методів оброблення природномовних даних шляхом спрощення публікації використаних вхідних даних.

Розглянемо приклади наявних природних і генерованих корпусів.

Традиційно, природні текстові корпуси є популярнішими для тестування нових методів оброблення природномовних даних, ніж генеровані. А якщо використання великих корпусів для тестування є недоцільним через їх обсяг (наприклад, корпус української мови містить 6,6 Гб текстів [3]), то використовують лише їх певні підмножини чи значно менші корпуси. Наприклад, популярним корпусом для тестування ефективності методів кластеризації та класифікації є анотований корпус новин Reuters-21578 [4].

Також популярним підходом у тестуванні методів оброблення природної мови є використання корпусів, генерованих на основі природних текстів. Прикладом є метод генерування корпусу для тестування виправлення граматичних помилок, описаний у [5]. Як вхідні дані цей метод використовує природномовні тексти з сайту "Вікіпедія", а потім перетворює їх, виходячи з особливостей задачі виправлення граматичних помилок. Однак застосування цього методу все одно призводить до залежності від великої кількості вхідних даних (тексти "Вікіпедії"), що можуть бути не описані та потребують збирання й зберігання.

Можна виправити такі недоліки, розробивши метод генерування текстового корпусу, що використовує мінімально можливу кількість вхідних даних – власне, розмір корпусу для генерування. Корпус, генерований цим методом, можна застосовувати для розв'язання таких задач:

– тестування швидкості оброблення природномовних даних відповідними методами;

– тестування коректності програмних реалізацій методів оброблення природної мови.

Найкращих результатів застосування такого методу можна досягнути, тестуючи методи оброблення природномовних даних, що не аналізують семантику текстів, прикладом яких є метод кластеризації текстів k-means (k-середніх) [6], який використовує лише статистичні дані про корпуси.

Постановка задачі

Метою роботи є підвищити ефективність тестування програмних реалізацій методів оброблення природномовних текстових даних за критерієм кількості знайдених дефектів, розробивши новий метод генерування корпусів

текстових даних задля їх використання як вхідних даних для тестування.

Вимоги до розроблюваного методу генерування корпусів текстових даних

Першою вимогою до розроблюваного методу є використання як вхідних даних лише розміру корпусу для генерування. Його можна задати однією з характеристик:

– кількістю текстів N_{docs};

– кількістю унікальних слів (термів) N_{terms};

– загальною кількістю слів (термів) NT_{terms}.

Перехід від однієї характеристики до іншої можливий за умови визначення двох додаткових функцій:

– $f(x)$ для опису переходу від кількості унікальних термів до кількості текстів. Тоді $f(N_{terms}) = N_{docs}$, а $f^{-1}(N_{docs}) = N_{terms}$;

– $g(x)$ для обчислення кількості входжень терму x у корпус. Тоді $\sum_{i=1}^{N_{terms}} g(x_i) = TN_{terms}$.

На практиці використання кількості унікальних слів (або загальної кількості слів) є кращим, оскільки повніше описує розмір корпусу. Для розроблюваного методу вирішено використовувати N_{terms} у поєднані з фіксованою функцією обчислення кількості входжень термів.

Другою вимогою є детермінованість або повторюваність отримуваних результатів. Детермінованість означає, що описаний метод має генерувати однакові корпуси під час кожного запуску з однаковим набором вхідних даних. Тож використовувати генератори випадкових чисел або інші джерела нестабільності результатів під час реалізації цього методу небажано. Дотримання цієї вимоги призводить до повторюваності результатів використання вказаного методу на практиці в різних задачах оброблення природної мови.

Третьою вимогою є те, що генерований корпус повинен мати складну структуру взаємозв'язків текстів. Очевидно, що метод, який генерує корпус із 10 однакових текстів, має обмежене застосування. Наприклад, при використанні такого корпусу в задачах кластеризації чи класифікації всі тексти складатимуть один кластер або клас. Виконати цю вимогу можливо, розподіливши різні терми в різні тексти генерованого корпусу, бажано в різних кількостях. Тож описана вище функція $g(x)$ має повертати вектор розмірністю N_{docs}, де на позиції i зазначають кількість входжень терму x у текст із номером i.

Детермінований метод генерування корпусів текстових даних

На основі сформованих вимог у межах даного дослідження запропоновано детермінований метод генерування корпусів текстових даних CorDeGen.

Метод включає такі етапи:

– приймання на вхід N_{terms};

– обчислення N_{docs} за допомогою функції $f(x)$;

– отримання вектора $\overline{tf}$ для кожного терму i, що містить кількість входжень терму до текстів, за допомогою обчислення функції $g(x)$;

– запис до кожного тексту терму i, базуючись на обчисленій кількості входжень.

За результатами проведених експериментів, як функцію $f(x)$ запропоновано використати функцію $\sqrt[4]{x}$.

Значення функції $g(x)$ запропоновано знаходити за таким алгоритмом:

– для терму з індексом i визначити його загальну кількість входжень у корпус за формулою $N_i = N_{docs}(i \bmod N_{docs} + 1)$;

– визначити індекс "центрального" тексту цього терму як $c_i = \bmod N_{docs}$;

– визначити границю діапазону входжень терму як $r = \left\lfloor \frac{N_{docs}}{5} \right\rfloor + 1$;

– у "центральний" текст записати $\frac{2}{2r+2} N_i$ входжень терму i;

– до решти текстів із діапазону $(c_i - r \ldots c_i + r)$ записати $\frac{1}{2r+2} N_i$ входжень терму i.

Водночас діапазон $(c_i - r \ldots c_i + r)$ є замкненим відносно корпусу, тобто, якщо $c_i - r < 0$ (за нумерації з нуля текстів у корпусі), то як індекс має бути взято значення $N_{docs}|c_i - r|$. Аналогічно реалізуємо обчислення для випадку, коли $c_i + r \geq N_{docs}$.

Отже, значення j-го елементу вектора $\overline{tf}$ для терму i можна подати формулою

$$tf_{ij} = \begin{cases} 0, j \notin (c_i - r \ldots c_i + r) \\ \frac{1}{2r+2} N_i, j \in (c_i - r \ldots c_i + r), j \neq c_i \\ \frac{2}{2r+2} N_i, j = c_i. \end{cases}$$

Алгоритм, що реалізує розроблений метод, є таким:

1. Обчислити параметри N_{docs} і r як

$$N_{docs} = \left\lfloor \sqrt[4]{N_{terms}} \right\rfloor,$$

$$r = \left\lfloor \frac{N_{docs}}{5} \right\rfloor + 1.$$

Для i від 0 до N_{terms} (не включно):

a) обчислити строкове значення терму, що буде занесене до текстів, за допомогою запису числа i в шістнадцятковой системі числення;

b) обчислити загальну кількість входжень терму i до корпусу як

$$N_i = N_{docs}(i \bmod N_{docs} + 1).$$

c) обчислити індекси текстів, до яких входитиме терм i як

$$c_i = i \bmod N_{docs},$$

$$i_{range} = (c_i - r \ldots c_i + r);$$

текст з індексом c_i записати i-й терм $\frac{2}{2r+2} N_i$ разів. До всіх інших текстів, індекси яких належать діапазону i_{range}, записати по $\frac{1}{2r+2} N_i$ входжень.

Особливості програмної реалізації методу

Запропонований детермінований метод генерування корпусів текстових даних реалізовано мовою програмування C# на платформі .NET Core 3.1 [7]. Алгоритм генерування корпусу та функції $f(x)$ і $g(x)$ є розділеними: об'єкт класу CorpusGenerator приймає екземпляр об'єкта DynamicPlugin, що є обгорткою над об'єктом будь-якого типу. Єдиним обмеженням на тип, що приймає DynamicPlugin, є необхідність надання двох публічних методів GetTextCount ($f(x)$) і GetTermDistribution ($g(x)$). Такий підхід дав змогу реалізувати динамічне завантаження вихідних кодів функцій $f(x)$ і $g(x)$ з окремого користувацького .cs файлу, завдяки чому експерименти щодо підбору вищеназваних функцій були виконані без перекомпіляції всього програмного забезпечення.

Розроблене програмне забезпечення працює за таким алгоритмом:

1. Прийняти три вхідні параметри: кількість термів (N_{terms}), шлях до файлу з реалізацією функцій $f(x)$ і $g(x)$, шлях до директорії для збереження результатів.

2. Динамічно завантажити та скомпілювати файл із реалізацією функцій $f(x)$ і $g(x)$, отримавши об'єкт класу DynamicPlugin. За цю процедуру відповідає клас DynamicPluginLoader, що для компіляції використовує компілятор Roslyn [8].

3. Створити екземпляр класу CorpusGenerator, передавши кількість термів й об'єкт класу DynamicPlugin.

4. Згенерувати корпус.

5. Зберегти отриманий корпус у вказаній директорії як текстові файли (один файл – один текст).

Щоб уникнути дублювання обчислення параметра N_i кожним із динамічних файлів, потрібно, щоб функція GetTermDistribution повертала лише відносне число розподілу термів, а не абсолютну кількість входжень. При обчисленні tf_{ij} ця функція повертатиме вектор виду $\bar{x} = (0, 0, \ldots, 1, 2, 1, \ldots, 0)$. Далі кожен елемент цього вектора слід помножити на N_i і розділити на $\sum x_i$ (яка дорівнює $2r + 2$ для стандартної функції $g(x)$).

Оцінювання швидкодії детермінованого методу генерування корпусів текстових даних

Для оцінювання використано бібліотеку BenchmarkDotNet, яка є де-факто промисловим стандартом галузі для платформи .NET [9].

Тестування виконано на ПК з апаратними характеристиками: Intel Core i7-9750H CPU 2.60GHz, 1 CPU, 12 логічних і 6 фізичних ядер. Результати подано в табл. 1.

***Таблиця* 1.** Час генерування корпусу

N_{terms}	Mean	Error	StdDev
100	48.89 μs	0.964 μs	2.033 μs
500	273.00 μs	3.185 μs	2.979 μs
2500	3.47 ms	47.277 μs	44.223 μs
12500	35.73 ms	224.062 μs	187.102 μs
62500	349.24 ms	2.42 ms	2.26 ms
312500	4.30 s	84.11 ms	128.44 ms

Примітка. Mean – середнє арифметичне всіх вимірювань. Error – половина довірчого інтервалу в 99,9 %. StdDev – стандартне відхилення всіх вимірювань.

За збільшення N_{terms} у 5 разів час, необхідний для генерування корпусу, збільшується в 10–12 разів, що приблизно відповідає складності $O(N^{1,5})$. Це майже відповідає мінімальній теоретичній складності, що дорівнює $O(N^{1,25})$ (кожен терм із колекції розміром N записують до колекції текстів розміром $O(N^{0,25})$).

Застосування методу CorDeGen для задач оброблення природної мови

Для демонстрації використання корпусу для тестування коректності програмних реалізацій методів оброблення природної мови, генерованого за допомогою запропонованого методу, у рамках дослідження виконано кластеризацію корпусу методом k-means (k-середніх).

Метод k-means – це один із найпопулярніших методів кластеризації, вперше запропонований у 1950-х роках [6]. В основу цього методу покладено ідею мінімізації сумарного квадратичного відхилення точок кластерів від їх центроїдів. У разі застосування цього методу до корпусу текстів, його подають як набір векторів (VSM, Vector Space Model [10]).

Щоб оцінити коректність програмної реалізації певного методу розв'язання задачі оброблення текстів, складених природною мовою, потрібно визначити очікуваний результат при застосуванні цього методу до генерованого корпусу. Для визначення такого очікуваного результату в контексті методу k-means розглянемо тексти з номерами N, $N-1$, $N+1$.

Нехай текст N складається з множини термів x, текст $N-1$ – із множини термів y, а текст $N+1$ – із множини термів z. З огляду на особливості запропонованого методу можна визначити такі відмінності між цими множинами:

– терм i, для якого текст N є центральним, у множині x трапляється вдвічі частіше, ніж у множинах y і z. Аналогічно для термів, центральними текстами яких були тексти $N-1$ і $N+1$;

– якщо для j-го терму $c_j + r = N - 1$, то терм відсутній у множинах x і z;

– якщо для k-го терму $c_k + r = N + 1$, то терм відсутній у множинах x і y.

Тож якщо між текстами з номерами M й O знаходиться щонайменше $2r$ текстів, то ці тексти не матимуть жодного спільного терму, проте між ними існуватиме ланцюжок "переходів" від термів першого тексту до термів другого – тим більший, чим більшою буде різниця між номерами текстів. У контексті кластеризації методом k-means це означає, що найбільш зрозумілим й очікуваним результатом буде рівномірний розподіл генерованого корпусу між k кластерами, а номери текстів, що потрапили до одного кластера, мають утворювати неперервний інтервал. Наприклад, тексти з номерами 0–4 потрапляють до кластера № 1, а з номерами 5–9 – до кластера № 2 (за $k = 2$, $N_{docs} = 10$).

У межах цього дослідження використано програмні реалізації векторизації текстів і методу k-means, надані бібліотекою Microsoft.ML [11]. Ця бібліотека як реалізацію методу k-means за замовчуванням використовує його різновид під назвою Yinyang k-means [12]. Він є поліпшеною версією методу k-means із погляду швидкості кластеризації та вибору початкових центроїдів (замість випадкового вибору використовують спеціальну процедуру).

Після кластеризації генерованого корпусу розміром $N_{terms} = 130321 (N_{docs} = 19)$ при двох кластерах отримано результати, зображені на рис. 1.

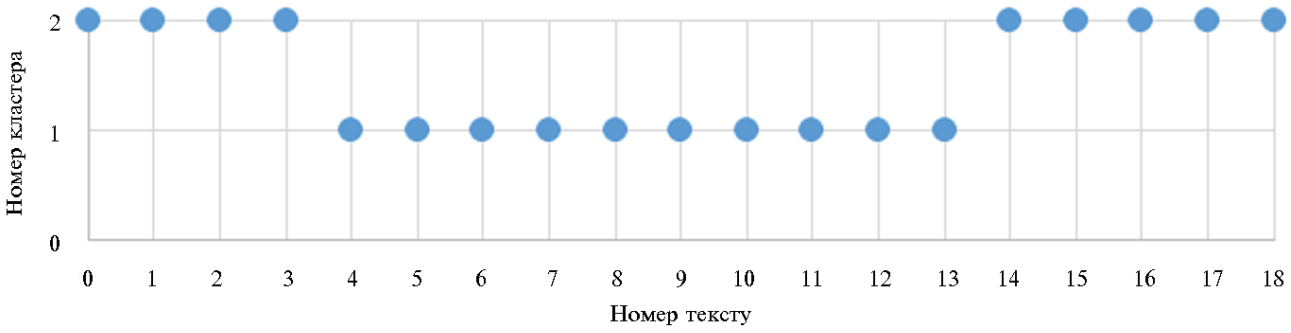

Рис. 1. Результати кластеризації генерованого корпусу методом k-means

Як бачимо з рис. 1, отримано очікуваний результат – рівномірний розподіл за кількістю текстів (10 і 9 текстів у кластерах відповідно), а номери текстів усередині кластера складають неперервний діапазон (для кластера № 2 потрібно враховувати, що CorDeGen є замкнутим відносно множини текстів, тобто діапазони 14–18 і 0–3 – це один неперервний діапазон для алгоритму).

Звісно, на отриманий результат впливають особливості самого методу k-means: він гарантує досягнення не глобального, а лише локального мінімуму сумарного квадратичного відхилення. Тому щоб досягти статистично вагомих результатів, генерований корпус кластеризовано декілька разів. Усі результати були подібними до тих, що наведені на рис. 1 – рівномірний розподіл за кластерами з неперервними інтервалами. Серед відмінностей отримано лише інверсію розмірів кластерів (з 10 і 9 на 9 і 10).

Щоб показати можливість виявлення програмної помилки під час реалізації методу, у програмну реалізацію методу k-means було навмисно внесено помилку – вектор, що відповідає тексту № 0, програмна реалізація замінює на вектор, що складається з одиниць. Результати кластеризації з помилкою в реалізації наведено на рис. 2.

Цей результат істотно відрізняється від очікуваного – усі тексти, окрім тексту № 0 (саме в який і було внесено помилку), віднесено до кластера № 1, і кластер № 2 складається з одного тексту № 0.

Тож корпус, що генерується застосуванням запропонованого детермінованого методу, дійсно можна використовувати для тестування коректності програмних реалізацій методів оброблення природномовних текстових даних. Отже, завдяки запропонованому алгоритму, можливо апріорі визначити еталонний, очікуваний результат, порівняння з яким дає змогу робити висновки про коректність реалізованого програмного забезпечення.

Висновки

Дослідження довело доцільність розроблення нових методів генерування текстових корпусів задля їх подальшого використання при тестуванні методів оброблення природної мови (для кластеризації, класифікації тощо). Запропонований детермінований метод генерування корпусів текстових даних (CorDeGen) залежить лише від необхідної кількості унікальних термів і створює корпуси складної структури. Загалом текстові корпуси, сформовані з використанням цього методу, можна використовувати як вхідні дані при тестуванні швидкості методів оброблення природномовних даних, а також при тестуванні коректності програмних реалізацій таких методів.

Алгоритм, що реалізує цей метод, має теоретичну обчислювальну складність $O(N^{1,25})$. На практиці, після замірів швидкості генерування корпусу виконаною програмною реалізацією, отримано складність близько $O(N^{1,5})$. Середній час генерування корпусу, що містить 312 500 унікальних термів, склав 4,3 секунди.

Як приклад застосування розробленого методу для тестування коректності програмної реалізації у цій роботі описано його використання для методу кластеризації k-means. Вказаний приклад довів ефективність методу CorDeGen, оскільки результат, отриманий програмною реалізацією з внесеною помилкою, відрізняється від очікуваного (визначеного на основі властивостей методів k-means і CorDeGen).

Попри те, що запропонований метод CorDeGen розроблено як універсальний, можна виділити такі його обмеження:

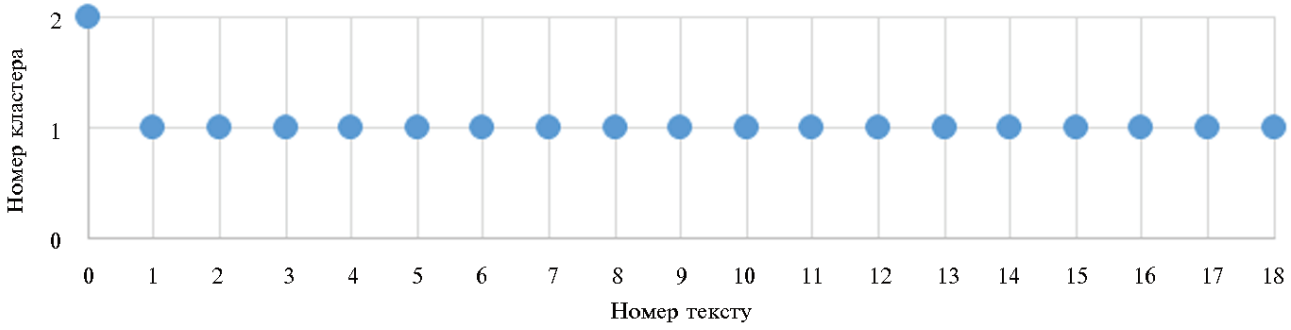

Рис. 2. Результат кластеризації генерованого корпусу методом k-means, програмна реалізація якого містить помилку

– оскільки CorDeGen використовує шістнадцяткові числа як терми, методи оброблення природної мови, що попередньо обробляють тексти, можуть видаляти частину термів. Наприклад, число *"A"* може бути видалено як стопслово (артикль в англійській мові), змінюючи розподіл термів між текстами й ускладнюючи визначення очікуваного результату при тестуванні;

– обмежена можливість використання CorDeGen разом із методами оброблення природної мови, що включають семантичний аналіз, оскільки семантика (сенс) відсутня в більшості згенерованих термів. Це також пов'язано з використанням шістнадцяткових чисел як термів, адже, наприклад, терм *"A4E"* не має сенсу в жодній із природних мов.

Подальші дослідження доцільно проводити в таких напрямах:

– оформлення програмної реалізації як бібліотек для різних мов програмування;

– застосування розробленого методу для тестування програмних реалізацій різних методів оброблення природної мови з описом очікуваних результатів.

Вихідні коди розробленого програмного забезпечення доступні за посиланням: https://github.com/yakivyusin/CorDeGen.

References

[1] N.S. Dash and S. Arulmozi, *History, Features, and Typology of Language Corpora*, Singapore: Springer, 2018, doi: 10.1007/978-981-10-7458-5.

[2] W3-Corpora Project. (2012, February 5). *Introduction: Corpus Linguistics* [Online]. Available: https://www1.essex.ac.uk/linguistics/external/clmt/w3c/corpus_ling/content/introduction2.html.

[3] Library "Chtyvo". (2017, October 13). *Corpus of Ukrainian language* [Online]. Available: http://korpus.org.ua.

[4] The UCI KDD Archive. (1999, February 16). *Reuters-21578 Text Categorization Collection* [Online]. Available: https://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html.

[5] J. Lichtarge, C. Alberti, S. Kumar, N. Shazeer, N. Parmar, and S. Tong, "Corpora Generation for Grammatical Error Correction," in *Proc. 2019 Conf. North American Chapter Association for Computational Linguistics: Human Language Technologies*, Minneapolis, MN, 2019, pp. 3291–3301, doi: 10.18653/v1/N19-1333.

[6] J.B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc Fifth Berkeley Symp. Mathematical Statistics and Probability*, 1967, pp. 281–297.

[7] Microsoft Ignite. (2019, April 12). *What's new in .NET Core 3.1* [Online]. Available: https://docs.microsoft.com/en-us/dotnet/core/whats-new/dotnet-core-3-1.

[8] Roslyn. (2015, January 14). *The Roslyn .NET compiler* [Online]. Available: https://github.com/dotnet/roslyn.

[9] BenchmarkDotNet. (2018, January 14). *Overview | BenchmarkDotNet* [Online]. Available: https://benchmarkdotnet.org/articles/overview.html.

[10] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, 1975, doi: 10.1145/361219.361220.

[11] Microsoft. (2018, May 7). *ML.NET* [Online]. Available: https://dotnet.microsoft.com/apps/machinelearning-ai/ml-dotnet.

[12] Y. Ding, Y. Zhao, X. Shen, M. Musuvathi, and T. Mytkowicz, "Yinyang K-Means: A Drop-In Replacement of the Classic K-Means with Consistent Speedup," in *Proc. 32nd Int. Conf. Machine Learning*, Lille, France, 2015, pp. 579–587.

Я.О. Юсин, Т.М. Заболотняя

ГЕНЕРАЦИЯ КОРПУСОВ ТЕКСТОВЫХ ДАННЫХ НА ОСНОВЕ ДЕТЕРМИНИРОВАННОГО МЕТОДА

Проблематика. Решение множества задач в области обработки естественного языка предусматривает использование корпусов текстовых данных, что делает актуальной подготовку последних. Их формирование на основе подлинных текстов требует временных затрат и не всегда целесообразно. Поэтому популярность приобретает автоматизированная генерации корпусов на основе различных методов и алгоритмов, что значительно упрощает подготовку экспериментальных данных.

Цель исследования. Увеличить количество обнаруживаемых дефектов во время тестирования программных реализаций методов обработки текстовых данных на естественном языке, путем разработки нового метода генерации корпусов текстовых данных для их использования в качестве входных данных для тестирования.

Методика реализации. Предложен детерминированный метод генерации корпусов текстовых данных CorDeGen (Corpora Deterministic Generation), который удовлетворяет следующим требованиям: детерминированность, зависимость на входе только от желаемого количества термов в генерируемом корпусе, обеспечение генерирования корпуса нетривиальной структуры. На основе предложенного метода разработан соответствующий алгоритм, а также осуществлена программная реализация на платформе .NET (язык программирования – C#). С помощью этой программной реализации выполнено оценивание быстродействия и эффективности разработанного метода.

Результаты исследования. Оценивание быстродействия разработанного метода CorDeGen показало степенную зависимость времени генерирования корпуса от количества термов с показателем степени близким к 1,5. В рамках исследования целесообразность использования разработанного метода для тестирования корректности программных реализаций показана на примере тестирования метода кластеризации k-средних.

Выводы. Тестирование разработанного детерминированного метода генерации корпусов текстовых данных показало его эффективность при тестировании других методов обработки естественноязычных данных, таких как кластеризация, вместо природных корпусов.

Ключевые слова: корпус текстовых данных; генерация корпусов; обработка естественноязычных данных; кластеризация данных; метод k-means; метод k-средних.

Ya.O. Yusyn, T.M. Zabolotnia

TEXT DATA CORPORA GENERATION ON THE BASIS OF THE DETERMINISTIC METHOD

Background. The solution to many problems in the field of natural language processing involves the use of corpora of text data, which makes the issue of preparing such corpora topical. At the same time, the formation of corpora based on natural texts is time-consuming and not always expedient. Therefore, an automated generation of corpora based on various methods and algorithms is gaining popularity, which greatly simplifies the preparation of experimental data.

Objective. The purpose of the paper is to increase the number of detected defects during testing of software implementations of methods for processing natural text data by developing a new method for generating text data corpora to be used as input for testing.

Methods. The deterministic method of text data corpora generation, named CorDeGen, is proposed, which satisfies the following requirements: determinism, dependence on input only from the desired number of terms in the generated corpus, as well as the non-trivial structure of the generated corpus. Based on the proposed method, the algorithm has been developed that implements it, as well as a software implementation on the .NET platform (programming language – C#). The evaluation of the speed and efficiency of the developed method has been done based on the developed software.

Results. The performed speed evaluation of the developed CorDeGen method showed a power-law dependence of the time of generating the corpus on the number of terms (input parameter), with a degree of about 1.5. In this study, the feasibility of using the developed method to test the correctness of software implementations is shown by the example of testing the k-means clustering method.

Conclusions. Testing of the developed deterministic method of text data corpora generation has shown the effectiveness of using this method in the testing of other natural language processing tasks, such as clustering, instead of natural corpora.

Keywords: corpus of text data; corpora generation; natural language processing; data clustering; k-means method.

Рекомендована Радою
факультету прикладної математики
КПІ ім. Ігоря Сікорського

Надійшла до редакції
27 вересня 2021 року

Прийнята до публікації
7 грудня року