ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА КОМП'ЮТЕРНА ТЕХНІКА

https://doi.org/10.31649/1997-9266-2024-172-1-45-50

УДК 004.02

А. І. Синько[1]

ВИЯВЛЕННЯ ПОДІБНОСТІ МІЖ ТЕКСТАМИ ДОПИСІВ ВІРТУАЛЬНИХ СПІЛЬНОТ ДЛЯ ФОРМУВАННЯ ДОКУМЕНТАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

[1]Національний університет «Львівська політехніка»

Галузь інформаційних технологій складається з двох суттєво різних частин: виробництво інформаційної техніки (машин, обладнання, програм тощо) і виробництво безпосередньо інформації, яка повинна бути задокументованою. На сьогодні попит на розробку програмного забезпечення є рекордно високим і навіть перевищує пропозицію на ринку. Важливою рисою програмного забезпечення є наявність належної супровідної документації, яка є потрібною як для розробників, так і для кінцевих споживачів. Інформаційними джерелами для формування документації програмного забезпечення можуть бути віртуальні спільноти, які є найвідвідуванішими ресурсами серед користувачів мережі Інтернет. Переваги використання віртуальних спільнот перелічені у роботі. Однією з характеристик документації є наявність унікального інформаційного наповнення, для виконання якого, після завантаження дописів з відібраних експертом віртуальних спільнот до сховища даних, необхідно перевірити їхній вміст. В результаті чого розроблено підхід до виявлення подібності, який відображає косинусоїдну подібність між всіма наявними дописами. Проведене дослідження показало, що більшість дописів містять унікальний контент, але деякі можуть мати подібні тексти. Перевагою застосування підходу до виявлення подібності є те, що після його виконання серед усіх попарно порівнювальних дописів можна визначити позиції пар дописів, значення мір яких зацікавлять дослідника та дозволять провести аналіз за різними методами. Досліджено випадки подібності тестів дописів та описано дії щодо їхнього вирішення, одним з яких є об'єднання подібних дописів та збереження всіх коментарів. Рекомендовано для дописів, що отримали високе значення міри подібності завдяки описаному підходу, надалі застосувати метод N-грам, який дозволить виявити ті частини текстів, що є різними для подальшого прийняття рішень.

Ключові слова: віртуальна спільнота, документація, програмне забезпечення, косинусоїдна подібність.

Вступ

В умовах глобальної комп'ютеризації та значного розвитку інформаційного суспільства спостерігається збільшення попиту на розробку програмних продуктів та швидкий розвиток індустрії комунікаційних технологій в цілому. Разом з тим зростає кількість інформаційних ресурсів у відкритому доступі, що містять корисний контент про програмне забезпечення (ПЗ). Одним з видів таких ресурсів є віртуальні спільноти (ВС), де користувачі спілкуються, тим самим обмінюючись своїми знаннями та досвідом, що може бути корисним для формування контенту документації. Відомо, багато українських ІТ-компаній основну увагу приділяють саме розробці ПЗ, але не супроводжують його якісною технічною документацією [1]. Це зумовлено тим, що формування документації програмного забезпечення (ДПЗ) потребує додаткових затрат (фінансових, часових і ресурсних), що не кожна компанія може собі дозволити. В результаті чого ДПЗ є відсутньою або неякісною. Неякісна ДПЗ впливає на якість наданих ІТ-послуг через неправильне інформування та якість створеного продукту. Все це викликає незадоволення як зі сторони замовника, так і користувачів ПЗ, негативний досвід яких розповсюджується і послаблює репутацію фірм [1].

Переваги використання ВС для формування ДПЗ можна подати з таких поглядів:

1. Для кінцевих споживачів, користувачів: отримання цільової інформації — вузькоспеціалізовані спільноти за певною тематикою; перегляд коментарів від інших користувачів; необмежений доступ до матеріалів; відсутність мовного бар'єру; доступ з будь-якої країни світу.

© А. І. Синько, 2024

2. Для взаємодії між учасниками (розробниками) та користувачами ПЗ: проведення аналізу та формування необхідної статистики, відстеження вподобань та відгуків користувачів; комунікація на відстані; взаємодія в реальному часі; оперативне надання відповіді.

Проаналізувавши переваги використання ВС різними зацікавленими особами, для отримання інформації про ПЗ виявлено, що спільноти є хорошим джерелом для формування інформаційного наповнення ДПЗ.

Для збереження інформації та подальшого її оброблення з певної предметної області застосовують сховища даних. Джерела інформації (віртуальні спільноти) зазвичай є однотипними, подібними але при цьому можуть мати різну структуру. Тому для роботи з даними потрібно їх привести до єдиного вигляду та надалі завантажити до сховище даних. Проаналізувавши методи збору даних (федералізація, тиражування та консолідація) з різнорідних джерел, для формування документації вибрано метод консолідації даних, адже саме він дає можливість об'єднувати дані з різних джерел в одному звіті [2]. На основі цього введено поняття — сховище консолідованих даних (СКД), що містить набір структурованих, представлених в одному вигляді даних завантажених з ВС, за допомогою методу консолідації. Після цього отриману інформацію можна опрацьовувати для формування ДПЗ як готового інформаційного продукту.

Важливою характеристикою ДПЗ є наявність унікального інформаційного наповнення, однією з властивостей якої є компактність представлення даних. Відомо, що користувачі можуть бути зареєстрованими на різних ВС та публікувати подібну або однакову інформацію у дописах, яка є корисною для формування інформаційного наповнення ДПЗ. З метою забезпечення характеристики унікальності контенту ДПЗ необхідно після процедури завантаження дописів перевіряти їхній контент. Цю процедуру можна виконати, застосувавши різні методи та моделі, що дозволяють встановити унікальність тексту, одним з таких методів є виявлення подібності між текстами дописів.

На сьогодні ефективним способом вирішення питання щодо встановлення унікальності тексту є технологія розпізнавання образів [3]. Розпізнавання образів трактують як віднесення даних, що є на вході, до певного класу із застосуванням важливих ознак і властивостей, притаманних цим даним [3]. Класом образів є певна їхня категорія, що утворюється разом зі спільними властивостями для кожного елемента. Основною задачею стає віднесення визначених образів до певних класів. Образ містить опис елементів всіх що є в класі образів. За наявності великої кількості даних застосовують автоматичне розпізнавання образів, що має практичне застосування в галузі штучного інтелекту.

Перевагою застосування розпізнавання образів для перевірки вмісту дописів ВС, завантажених до СКД, є надання якісної оцінки унікальності даних.

Основними етапами цієї технології є [3]: пошук тематично близьких по контенту (подібних) текстів, які отримано з відібраних тематичних ВС та завантажено у СКД для подальшої роботи; оброблення вхідного інформаційного наповнення допису; встановлення відсотку унікальності вмісту дописів.

Отже, враховуючи потребу у формуванні ДПЗ, інформаційним наповненням якої можуть бути дописи ВС, виникає потреба у перевірці їхнього вмісту за допомогою встановлення міри подібності для уникнення дублювання даних.

Метою роботи є розробка підходу до виявлення подібності між текстами дописів віртуальних спільнот для формування унікального інформаційного наповнення документації програмного забезпечення.

Основна частина

Під час роботи з великими обсягами даних, які знаходяться у ВС та можуть бути корисними для формування ДПЗ, виникає потреба у встановленні міри подібності між дописами. Адже подібні дописи можуть містити однакові дані, плагіат, спам тощо.

Встановлення подібності між текстами дописів за запропонованим підходом відбувається таким чином:

1. Нормалізація текстів дописів за допомогою морфологічного аналізу — лематизації [4], [5].

2. Формування спільного словника для всіх дописів.

3. Застосування показника TF-IDF [6] для кожного тексту допису, який відображає оцінку важливості слів у документі на основі частоти їхнього використання у тексті.

4. Перетворення кожного тексту допису у вектори [7]. Координати вектора утворюють слова тексту. Важливо врахувати, що значення кожного елементу вектора має відображати те саме слово у векторах для всіх документів. Всі вектори повинні бути приведені до єдиної довжини (розміру).

5. Обчислення значення косинусу подібності між векторами (див. (1)).

Для проведення аналізу тексту допису за різними методами попередньо необхідно його нормалізувати. Нормалізація — це метод, у якому кожне слово тексту приводиться до його словникової форми, щоб скоротити час пошуку. Існують різні способи нормалізації тексту, одним з яких є лематизація, що полягає у знаходженні леми слова залежно від його значення.

Нормалізація тексту полягає у виконанні таких кроків:

1. Стандартизація тексту полягає у розбитті тексту на лексеми (слова) та застосуванні токенизації (очищенні слів від апострофів, знаків пунктуації) (рис. 1).

2. Вилучання слів, які не беруться до уваги під час проведення лінгвістичного аналізу, адже вони не несуть важливу інформацію для тексту (займенники, артиклі, прийменники, сполучники) — стоп слова (з англ. stop words).

3. Застосування алгоритму лематизації для англомовного або україномовного тексту, що дозволить отримати леми (основну форму слів).

Результатом проходження цього алгоритму є нормалізація слів, що містяться в тексті:

• для англомовного слова — лема;

• для україномовного слова — множина лем (частини мови та відмінки слова у тексті).

Існують випадки, коли слова україномовного тексту можуть не отримати лему, наприклад англомовні слова (назви ПЗ тощо) або який немає у словнику (слова з помилками). В таких випадках слова належать до таких класів:

• «unclass» — для слів інших мов;

• «unknown» — для слів, яких немає у словнику.

```
def standardize(text: str):
    text = text.lower()            # приведення до нижнього регістру символів
    text = text.replace('\\n', ' ')
    text = re.sub(punct_pattern, ' ', text)
    text = " ".join([process(word) for word in text.split(' ') if word not in
                     stop_words])
    return text

return standardize
```

Рис. 1. Функція стандартизації тексту

Після того, як тексти дописів нормалізовано, необхідно сформувати спільний словник термінів та застосувати для кожного допису статистичний показник *TF–IDF*

$$TF - IDF = TF \cdot IDF = \frac{n_t}{\sum\limits_{i=1}^{k} n_i} \cdot \ln\left(\frac{n_c}{\sum\limits_{j=1}^{m} n_j}\right), \quad (1)$$

де n_t — число входжень *t*-го слова щодо тексту допису; n_i — загальна кількість слів у дописі; n_c — кількість дописів за якими сформовано словник; n_j — кількість дописів в яких виявлено *t*-те слово.

Для виявлення подібності векторів визначають відстань між ними за допомогою обчислення косинусу кута між ними, що представляється за допомогою скалярного добутку та норми кожного вектора. Косинус кута між двома векторами обчислюється так [7], [8]:

$$\cos(\theta) = \frac{Vector_1 \cdot Vector_2}{\|Vector_1\| \cdot \|Vector_2\|}, \quad (2)$$

де $Vector_1$ — координати першого вектора (тексту допису); $Vector_2$ — координати другого вектора; $\|Vector_1\|$ — норма першого допису; $\|Vector_2\|$ — норма другого допису.

Вектори вважаються подібними, якщо вони мають однаковий напрямок. При чому довжина тексту допису для оцінювання подібності не є важливою. У випадку виявлення подібності між двома текстами дописів значення косинусу кута може знаходитися в інтервалі [0, 1]. Якщо дописи є подібними, то значення косинусу кута буде дорівнювати одиниці. Якщо в текстах дописів не виявлено

подібності, тоді це значення буде дорівнювати нулю.

Реалізацію методу виявлення подібності між текстами дописів виконано за допомогою вбудованих компонент набору бібліотек NLTK, призначенням яких є символьна та статистична обробка природної мови, реалізованих мовою програмування Python.

Функція обчислення косинусоїдної подібності (get_similarity_func) між текстами дописів подана на рис. 2.

```
def get_similarity_func(vectorization, to_standard_func, pairwise_func):
    def compute_text_similarity(texts: list[str]):
        std_text = texts['Content'].apply(lambda x: to_standard_func(x))
        tfidf_tokens = vectorization.transform(std_text)
        return tfidf_tokens, pairwise_func(tfidf_tokens)

    return compute_text_similarity
```

Рис. 2. Функція обчислення косинусоїдної подібності між дописами

Результати обчислення подібності між текстами дописів показано у вигляді матриці (рис. 3), що відображає попарне порівняння дописів за допомогою обчислення косинусу кута між їхніми векторами за формулою (2). Вибірка, словник становить 180 дописів (на рис. 3 відображена частина дописів). Кількість слів становить 10 тисяч.

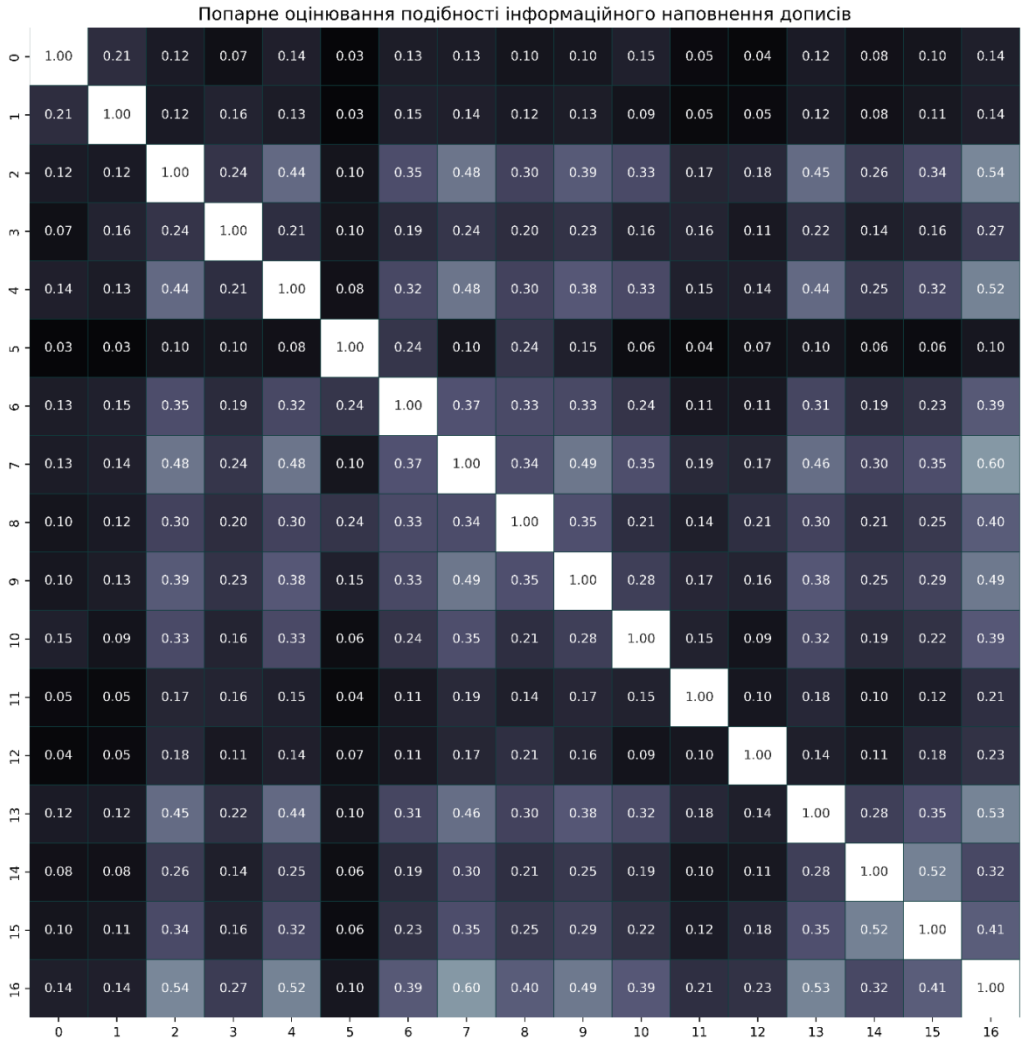

Рис. 3. Матриця попарної косинусоїдної подібності між дописами

Далі для отриманих значень коефіцієнтів кореляції між дописами необхідно здійснити перевірку міри подібності таким чином:

$$\mu_{similarity}(\cos\theta) \geq \alpha, \quad (3)$$

де α — показник, що визначає порогове значення для оцінки міри подібності тексту допису.

Для дописів, які задовольняють умові формули (3), потрібно сформувати список для подальшої перевірки. Якщо виявлено високе значення міри подібності (до прикладу, $\mu_{similarity}(\cos\theta) \geq 0{,}55$), то тексти дописів можуть:

1. Бути ідентичними (наприклад, автор розмістив однакові за змістом дописи на декількох ВС для охоплення більшої аудиторії — отримання більше відповідей і оцінок від користувачів на опублікований допис за короткий проміжок часу);

2. Бути семантично близькими за значеннями (наприклад, коли один автор публікує дописи на одну тематику);

3. Містити цитування іншого допису або плагіат (якщо цитування не вказано).

У першому випадку, якщо дописи є ідентичними тоді косинус кута буде дорівнювати 1. В інших двох випадках значення буде менше 1, але достатньо вагомим для здійснення подальшого аналізу вмісту дописів. Надалі, для дописів які задовольняють умові формули (3), доцільно провести аналіз за методами, які враховують довжину текстів дописів (адже один допис може бути частиною іншого). Одним з таких методів є метод N-грам [9], [10], який застосовують для виявлення плагіату. Метод N-грам працює повільніше, адже алгоритм виконання є складнішим, тому його слід застосовувати після методу виявлення подібності дописів. Перевагою застосування методу виявлення подібності є те, що після його виконання серед усіх попарно порівнювальних дописів можна визначити позиції пар дописів, які мають високе значення міри подібності. І в подальшому досліджувати за іншими алгоритмами саме ті пари дописів що цікавлять науковця.

Для подібних дописів ВС виникає потреба у їхньому об’єднанні таким чином:

– якщо дописи ідентичні, то вибираємо допис, що набрав більшу кількість реакцій;

– якщо один текст є доповненням іншого, тоді вибираємо текст, який має більшу довжину.

Також, у разі об’єднання дописів, необхідно перевіряти наявність коментарів до них. Якщо коментарі існують, тоді потрібно зберігати обидві «гілки» коментарів до дописів.

Як видно з результатів побудови матриці попарної косинусоїдної подібності між текстами дописів, більшість з них мають низьке значення міри подібності. Це говорить про те, що дописи містять унікальний контент. Але знайдено дописи, які мають достатнє значення міри подібності для їхньої додаткової перевірки перед тим, як вони можуть стати інформаційним наповненням ДПЗ. Проаналізувавши два дописи на рис. 3, значення міри подібності яких становить 0,6 (розташовано на перетині 7 і 16 позиції), виявлено, що вони опубліковані одним автором (що відповідає другому пункту у списку для перевірки дописів).

Висновки

Важливість наявності документації до програмного забезпечення, яка є потрібною як для розробників, так і для кінцевих споживачів, зумовлює потребу у її формуванні. На сьогодні віртуальні спільноти виступають хорошим джерелом інформації для формування інформаційного наповнення документації, переваги застосування яких зазначені у роботі. Однією з характеристик документації програмного забезпечення є забезпечення унікального інформаційного наповнення. Для виконання цього наповнення, після завантаження дописів з відібраних експертом віртуальних спільнот до сховища даних, необхідно перевірити та обчислити міру подібності між їхніми текстами. Результати дослідження показали, що більшість дописів містили унікальний контент, але деякі мали подібні тексти. Випадки подібності тестів дописів досліджені, а також описані дії щодо їхнього вирішення — об’єднання дописів та збереження усіх коментарів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

[1] О. В. Марковець, і А. І. Синько, «Формування якісної технічної документації до програмного забезпечення,» *Вісник Вінницького політехнічного інституту*, вип. 2 (155), с. 98-106, 2021. https://doi.org/10.31649/1997-9266-2021-155-2-98-106 .

[2] П. І. Жежнич, і О. О. Сопрунюк, «Консолідація відкритих інформаційних ресурсів в туристичній сфері,» *Комп’ютерні науки та інформаційні технології: Вісник Національного університету «Львівська політехніка»*, № 771, с. 3-11, 2013.

[3] Л. М. Колєчкіна, і О. П. Пухтєєва, «Розробка методу і алгоритму перевірки тексту на унікальність,» *Нові технології*, № 1-2, с. 58-62, 2013.

[4] К. К. Духновська, Я. А. Страшок, і П. В. Шило, «Інформаційна технологія для проведення лематизації і стемінгу в україномовних текстах,» *Прикладні системи та технології в інформаційному суспільстві, зб. тез VI Міжнародної науково-практичної конференції*, № 1, с. 119-127, 2013. Режим доступу: http://kist.ntu.edu.ua/konferencii/32_konf_2022.pdf#page=119 .

[5] D. Khyani, B. S. Siddhartha, N. M. Niveditha, and B.M. Divya, "An Interpretation of Lemmatization and Stemming in Natural Language Processing," *Journal of University of Shanghai for Science and Technology*, vol. 22 (10), pp. 350-357, 2020.

[6] A. Jalilifard, V. F. Carida, A. F. Mansando, R. S. Cristo, F. Penhorate, and C. Fonseca, "Semantic Sensitive TF-IDF to Determine Word Relevance in Documents," *Computing and Network Communications*, vol. 736, pp. 327-337, 2021. https://doi.org/10.1007/978-981-33-6987-0_27 .

[7] Ю. А. Кравченко, А. М. Мансур, і Ж. Х. Мохаммад, «Векторизация текста с использованием методов интеллектуального анализа данных,» *Известия ЮФУ*, № 2, с. 154-167, 2021. https://doi.org/10.18522/2311-3103-2021-2-154-167 .

[8] P. Kwangil, H. S. June, and K. Wooju, "A Methodology Combining Cosine Similarity with Classifier for Text Classification," *An International Journal Applied Artificial Intelligence*, vol. 34, pp. 396-411, 2020. https://doi.org/https://doi.org/10.1080/08839514.2020.1723868 .

[9] J. Awwalu, A. A. Bakar, and M. R. Yaakub, "Hybrid N-gram model using Naïve Bayes for classification of political sentiments on Twitter," *Neural Computing and Applications*, no. 31, pp. 9207-9220, 2019. https://doi.org/10.1007/s00521-019-04248-z .

[10] К. Т. Кузьма, «Інформаційна технологія оцінки рівня подібності рядків на основі методу N-грам,» *Вчені записки ТНУ імені В.І. Вернадського*, т. 31 (70), ч. 1, № 6, с. 96-99, 2020. https://doi.org/10.32838/TNU-2663-5941/2020.6-1/16 .

Рекомендована кафедрою автоматизації та інтелектуальних інформаційних технологій ВНТУ

Стаття надійшла до редакції 27.03.2023

Синько Анна Іванівна — аспірантка кафедри соціальних комунікацій та інформаційної діяльності, e-mail: anna.i.synko@lpnu.ua .

Національний університет «Львівська політехніка», Львів

A. I. Synko[1]

Detecting Similarity between the Texts of Posts of Virtual Communities for the Formation of Software Documentation

[1]Lviv Polytechnic National University

The field of information technologies consists of two significantly different parts: the production of information technologies (machines, equipment, programs, etc.) and the production of the information itself, which must be documented. Today, the demand for software development is at a record high and even exceeds the market supply. An important feature of software is the availability of proper accompanying documentation, which is necessary for both developers and end users. Information sources for the formation of software documentation can be virtual communities, which are the most visited resources among Internet users. The advantages of using virtual communities are given in the article. An important characteristic of software documentation is the provision of unique information content. To fulfill this requirement, it is necessary to check their content after uploading the publications to the data repository. It should be noted that virtual communities for the formation of software documentation should be thematic. As a result, a method was developed that displays the cosine similarity between all available posts. The research conducted showed that most of the posts contain unique content, but some may have similar texts. The advantage of using the similarity detection method is that after its execution among all pairwise comparison posts, the positions of pairs of posts can be determined. In the future, we will choose posts whose values will be of interest to the researcher and will allow us to conduct analysis using other methods. Next, cases of post similarity tests were investigated and actions to solve them were described, one of which is to joint similar posts and save all comments. It is recommended to use the N-gram method for posts that received a high value of the similarity measure using the cosine similarity method.

Keywords: virtual community, documentation, software, cosine similarity.

Synko Anna I. — Post-Graduate Student of the Chair of Social Communications and Information Activities, e-mail: anna.i.synko@lpnu.ua