Міністерство освіти і науки, молоді та спорту
Національний технічний університет України
“Київський політехнічний інститут”

Факультет інформатики та обчислювальної техніки
Кафедра автоматизованих систем обробки інформації та управління

ЛАБОРАТОРНА РОБОТА №3

з дисципліни “Комп’ютерна лінгвістика”
на тему: “Оцінка близькості документів методом шинглів”

Виконав: Турко М. В.
ст. гр. ІС-73
Перевірив: Фіногенов О. Д.

Київ 2020

Лабораторна робота №3

Мета: вивчення методів визначення подібних (близьких за текстом) документів, зокрема для знаходження плагіату.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

З появою мережі Інтернет та різким збільшенням інформації, однією з головних задач інформатизації суспільства є пошук необхідних даних, документів тощо. Постійне збільшення та накопичення даних призводить до необхідності розроблення алгоритмів порівняння текстів, зменшення множини текстів, на якій виконується пошук, визначення релевантності текстів та інші. На результатах пошуку подібних документів побудовані сучасні пошукові системи, які окрім безпосередньо видачі посилань враховують релевантність тексту (наприклад при визначення показника PageRank). Від місця знаходження посилання в пошуковій видачі залежить кількість відвідувачів сайту, що впливає на прибутки організації/ресурсу або її популярність. Поняття релевантності тексту та створення унікального контенту лежить, наприклад, в основі такого напрямку діяльності як рерайтінг.

Окремо стоїть задача пошуку подібних документів при визначенні плагіату та порушенні авторських прав на твір, програму, зображення тощо.

Метод «Шинглів»

В 1997 році Бродер [4] запропонував метод, оснований на представленні документа у вигляді послідовностей фіксованої довжини N, що складаються із сусідніх слів. При цьому на послідовності можуть накладатися обмеження, наприклад, слова повинні знаходитися в межах одного речення.

Такі послідовності в одних джерелах називають «шинглами» [5], в інших «N-грамами» [6].

Два документи вважаються подібними, якщо множини їх N-грам істотно перетинаються. Аналогічно можна вважати подібність двох речень, чи речення та тексту. Кількість N-грам для кожного документа є достатньо великою, тому потрібно використовувати різні способи зменшення їх множини.

Алгоритм не може використовуватися для визначення конкретних запозичень, так як будь-яка модифікація тексту при запозиченні приведе до того, що шингл зміниться, і факт плагіату не буде встановлено, що в свою чергу призведе до втрати повноти. Алгоритм може використовуватися для відбору документів на основі того, чи вважаються документ, що перевіряють і можливий документ-джерело подібними.

Метод «супершинглів» («мегашинглів»)

Модифікація алгоритму шинглів [7] заключається в тому, що кожний документ представляється 84 шинглами. Вибір із множини усіх шинглів відбувається за наступною схемою: для всіх шинглів документа розраховується значення 84 хеш функцій. Для кожної хеш функції обирається шингл з максимальним значенням хеш функції. Потім ці 84 шингла розбиваються на 6 груп по 14 шинглів. Такі групи називаються «супершинглами». Далі документ представляється всіма можливими попарними поєднаннями із 6 супершинглів, що називаються «мегашинглами». Кількість таких мегашинглів дорівнює 15. Два документи подібні за змістом, якщо у них співпадає хоча б один мегашингл.

Алгоритм являється нестійким при модифікації тексту при запозиченні, оскільки при модифікації тексту, відповідному шинглу зміниться також і

мегашингл. Також алгоритм не визначає плагіат у випадку малої подібності документів: відповідно статті [5], у випадку збігу документів на 96 відсотків, 2 супершингла збігаються з імовірністю 49 відсотків. Відповідно, запозичення одного абзацу в двох відносно великих документах буде недостатнім для визначення даним алгоритмом.

Описаний алгоритм можна використовувати для відбору документів для пошуку, якщо критерієм відбору є масштабні запозичення, для пошуку конкретних запозичень фрагментів алгоритм не є актуальним, оскільки мегашингл містить лише невелику підмножину шинглів документу, і, відповідно, не може містити інформацію про всі можливі запозичення. Також алгоритм не є найкращим через нестійкість до модифікацій: будь-яка модифікація тексту призведе до зміни відповідних шинглів, що призводить до низької повноти алгоритму.

Surrounding Context N-Grams

Алгоритм Surrounding Context N-Grams є модифікацією алгоритму шинглів [8]. В основу алгоритму береться припущення про те, що текст може модифікувати шляхом видалення одиничних слів, граматичних помилок, перестановки слів, і стандартний алгоритм шинглів не виявить співпадіння. Для цього збільшується число шинглів, що генеруються. Для кожної послідовності слів N береться деяке число M, менше N, та із послідовності слів N забирається M слів всіма можливими способами. Слова, що лишилися, сортуються в лексикографічному порядку і являються собою нові N-грами. Недоліком даного алгоритму є значний приріст числа шинглів, що призводить до значного сповільнення роботи програми. Також це призводить до збільшення кількості хибних спрацювань.

Описаний алгоритм є стійким до незначних модифікацій тексту (якщо модифікується не більше M слів підряд), і його можна використовувати для

пошуку конкретних запозичень, однак його не можна застосувати при модифікації більше M слів підряд. Задати константу M достатньо великою неможливо, оскільки це призведе до значного зниження точності алгоритму. Також може використовуватися для попереднього відбору документів за ознаками, аналогічними стандартному алгоритму шинглів.

Алгоритм шинглів

Згідно [9], щоб зробити висновок про те чи насправді два документи є нечіткими дублікатами, або один з документів включає в себе інший, використовуються математичні поняття схожості (resemblance) та входження (containment).

Схожість двох документів A та B – це число між 0 та 1, таке, що у разі схожості, близької до одиниці, два документи можна вважати нечіткими дублікатами. Так само входження документа A в B є число між 0 та 1, таке що при його наближенні до одиниці, можна зробити висновок, що документ B включає документ A. Для обчислення схожості та/або входження двох документів достатньо зберігати для кожного документа його короткий образ у декілька сотень байт. Такий короткий образ може бути ефективно побудований за час, прямо пропорційний довжині документа. Отже, схожість та входження для двох документів може бути обчислена за час, прямо пропорційний до розміру образу.

Текстовий документ розглядається як послідовність слів приведених до канонічної форми (токенів). Під канонічною формою мається на увазі видалення елементів форматування, HTML- тегів та знаків пунктуації. Тоді з кожним документом D можна пов’язати підмножину таких токенів S (D,w). Суміжні підпослідовності слів w, що входять в D, називаються «шинглами».

Отже, множиною «шинглів» для певного документа є набір всіх унікальних «шинглів», що входять до цього документа.

Для заданої довжини «шинглу», схожість r двох документів A та B можна визначити так [9]:

$$r(A,B) = \frac{|S(A) \cap S(B)|}{|S(A) \cup S(B)|}, \quad (3.1)$$

Тобто, це є відношення числа однакових для двох документів «шинглів» до загальної їх кількості. Входження документа A в документ B визначається як відношення числа однакових «шинглів» до числа «шинглів» A:

$$c(A,B) = \frac{|S(A) \cap S(B)|}{|S(A)|}, \quad (3.2)$$

При розпізнаванні нечітких дублікатів також використовується поняття відстані схожості (resemblance distance) документів, що визначається як:

$$d(A,B) = 1 - r(A,B), \quad (3.3)$$

Реалізація класичного алгоритму «шинглів» передбачає кілька основних етапів:

1. Канонізація тексту

На початковому етапі вхідні тексти приводяться до канонічного вигляду. Тобто з них видаляються усі знаки пунктуації, елементи форматування, зайві пробіли, HTML теги та стоп-слова.

2. Розбиття тексту на «шингли»

Цей етап передбачає розбиття канонізованого тексту на підрядки однакової довжини («шингли»). Найчастіше в якості кроку обираються символи або слова, і значно рідше – речення. Вибір кроку розміром у речення підходить лише для доволі великих документів і передбачає деякі зміни на

етапі канонізації текстів. «Шингли» повинні виділятися з тексту не один за одним, а накладатися. Мінімальна відстань (d), між двома сусідніми «шинглами» дорівнює 1 слово/символ і гарантує, що при розбитті тексту на «шингли» не буде жодних втрат інформації. При збільшенні відстані d зменшується розмір вибірки, що позитивно впливає на швидкодію алгоритму.

3. Знаходження контрольних сум «шинглів»

Після того як отримано множину «шинглів» для текстового документа, необхідно обчислити їхні контрольні суми. Для цього використовують хеш-функції, які забезпечують мінімальну кількість колізій. Найчастіше використовують криптографічні хеш-функції md5, crc, sha.

4. Побудова короткого образу документа

У класичному алгоритмі «шинглів» процес побудови образу передбачає відбір лише тих контрольних сум, які будуть використовуватися під час порівняння двох документів. Це можна зробити декількома способами. Можна відбирати задану кількість мінімальних за значенням контрольних сум. У цьому випадку отримаємо фіксовану за розміром вибірку (що є перевагою для коротких документів), але не можна буде судити про входження документів. Інший спосіб полягає в відборі контрольних сум, які діляться без залишку на деяке число з діапазону від 10 до 50. Такий метод називають відбором за модулем m (де $m \subset N$). Розмір вибірки в такому випадку буде пропорційний до розміру документа. Найпростішим є відбір абсолютно всіх «шинглів». Тоді для зберігання образу потрібно виділяти більше пам’яті, а порівняння двох документів займатиме більше часу.

5. Пошук однакових підпослідовностей

На останньому етапі відбувається порівняння двох образів шляхом визначення ступеня їх схожості і входження. Якщо значення r (c) перевищує заданий поріг, то документи можна вважати нечіткими дублікатами. Значення

порогу схожості визначається експериментально і здебільшого лежить у діапазоні 0,75–0,95.

Даний алгоритм можна модифікувати з метою зменшення чутливості до перестановки слів на етапі розбиття тексту на «шингли» згідно [9]. Якщо припустити, що на етапі розбиття тексту на «шингли» в якості кроку було обрано речення, а не слово, то в якості переставлених місцями фрагментів можна розглядати речення. Хоча прикладів таких змін, що не порушують викладену думку в тексті, підібрати значно складніше.

Задачу можна сформулювати так. Визначимо S1, як один з «шинглів», сформованих для деякого текстового документа, а S2 – як «шингл», отриманий з S1 шляхом зміни місцями двох довільних слів.

Тоді можна записати, що

$$S1 = \{w1,..., wi,..., wj,..., wn\} \quad (3.4)$$

$$S2 = \{w1,..., wj,..., wi,..., wn\}, \quad (3.5)$$

де wi, wj – переставлені місцями слова.

Нехай SH1 і SH2 хеш-суми обчислені для «шинглів» S1 та S2 за допомогою деякої хеш-функції hf. При порівнянні двох текстів, необхідно, щоб SH1 та SH2 набували однакових значень. Це означатиме, що перестановка слів не впливатиме на результати обчислення схожості r та входження c.

Тоді задача зменшення чутливості до зміни порядку слів зводиться до задоволення такого виразу:

$$SH1 = hf(S1) = hf(S2) = SH2 \quad (3.6)$$

або

$$SH1 = hf(w1,..., wi,..., wj,..., wn) = hf(w1,..., wj,..., wi,..., wn) = SH2 \quad (3.7)$$

,

Такого результату можна досягти на етапі розбиття канонізованого тексту на «шинглі», виконавши сортування за алфавітом усіх слів у межах кожного «шинглу». Це означає, що у разі будь-якої зміни порядку слів wi у межах «шинглу», він завжди матиме сталий вигляд:

$$S = \{wa, wb, wc, ..., wz\} \tag{3.8}$$

Отже, контрольні суми двох «шинглів» S1 та S2 (який отриманий з S1 методом перестановки слів) матимуть рівні значення, оскільки їх розраховують для однакових аргументів:

$$S1 = sort(w1, ..., wi, ..., wj, ..., wn) , \tag{3.9}$$

$$S2 = sort(w1, ..., wj, ..., wi, ..., wn) , \tag{3.10}$$

$$hf(S1) = hf(sort(w1, ..., wi, ..., wj, ..., wn)) , \tag{3.11}$$

$$hf(S2) = hf(sort(w1, ..., wj, ..., wi, ..., wn)) , \tag{3.12}$$

$$S1 = S2 = S \Rightarrow SH1 = hf(S1) = hf(S2) = SH2 \tag{3.13}$$

Важливою перевагою цього методу є відсутність будь-яких колізій, пов’язаних з операціями обчислення контрольних сум. Це означає, що хеш-суми двох «шинглів» S1 та S2 будуть рівними тільки в тому випадку, якщо вони складаються з однакового набору слів.

Результати вдосконалень алгоритму «шинглів» багато в чому залежать від граматики конкретної мови. Чим більш вільним є порядок слів у реченні, тим краще проявить себе запропонований метод зменшення чутливості.

Підтвердженням покращення роботи алгоритму можуть слугувати результати тестування на вибірці документів та їх нечітких дублікатів, створених шляхом перестановки слів, яка не змінює зміст речень.

При цьому чутливість алгоритму можна визначати як відношення

кількості правильно розпізнаних дублікатів до загальної кількості документів у вибірці:

$$Sens = \frac{N_{found}}{N_{existing}}, \tag{3.14}$$

де N_{found} – кількість правильно розпізнаних дублікатів; $N_{existing}$ – загальна кількість наявних дублікатів, отриманих внаслідок зміни порядку слів.

ХІД ВИКОНАННЯ

1. **Формування початкового датасету**

   Було взято по 50 анекдотів(частота класу в корпусі = 0,5) з двох категорій варіанту завдання (погода, політика). У таблиці 1.1. наведено вигляд файу .csv.

   Таблиця 1.1 - датасет

	Номер	Текст	Категорія
0	0	В хорошую погоду меня от окна даже за уши не о...	0
1	1	Как правило "левые" не правы.	1
2	2	Количество новостей про осадки превысило месяч...	0
3	3	Глупые народы, прикрывая свою глупость, воюют ...	1
4	4	Как объяснить знакомому испанцу, что если идёт...	0
...	...	...	...
95	95	Депутат, которого обвинили в получении второго...	1
96	96	А вот вы задумывались, что притопывая и подпры...	0
97	97	В день юбилея Ленина Зюганов заходит в бар и г...	1
98	98	Прогноз погоды - это такой вид новостей, к кот...	0
99	99	- Найди и принеси мне то, чего на белом свете,...	1

100 rows × 3 columns

2. **Дослідження корпусу документів**

По двом заданим категоріям проводиться кількісний аналіз слів по документам класу категорії. На рисунка 2.1-2.2 приводяться гістограми кількісних частот слів класу, включно зі стоп словами.

Рис. 2.1 - гістограма кількості вживаних слів у категорії “Погода”

Рис. 2.2 - гістограма кількості вживання слів у категорії “Політика”

На рисунках 2.3-2.4 показані гістограми кількості вживання слів двух категорій без стоп-слів.

Рис. 2.3 - гістограма кількості вживання слів у категорії “Погода”

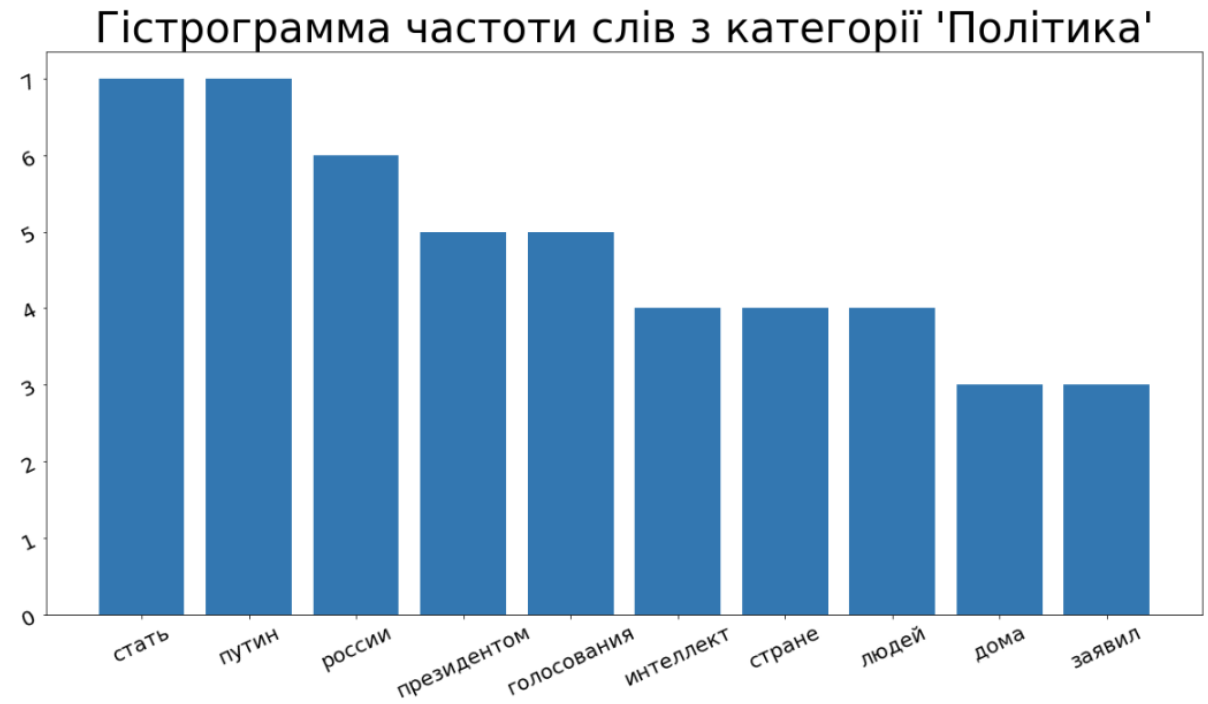

Рис. 2.4 - гістограма кількості вживання слів у категорії “Політика”

3. Дослідження впливу розміру навчальної вибірки до ймовірності потрапляння документу до классу

У таблиці 3.1 наведено значення ймовірностей відношення документу до классу відносно розміру початкової вибірки.

Таблиці 3.1 - ймовірності

	Навчальні вибірки	Ймовірність
0	Розмір: 10 x 10	0.600
1	Розмір: 20 x 20	0.625
2	Розмір: 30 x 30	0.725

На рисунку 3.1 продемонстровано графік оснований на таблиці 3.1.

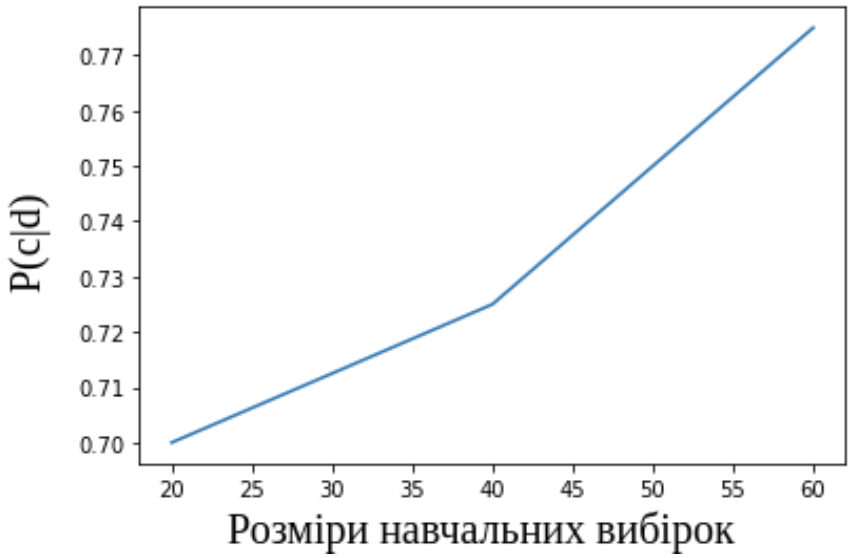

Рис. 3.1 - графік зростання ймовірності відносно збільшення розміру навчальної вибірки

Висновок:

Була досліджена задача визначення тональності тексту за допомогою використання наївного баєсіва класифікатора.

При дослідженні впливу розміру навчальної вибірки на визначення ймовірності потрапляння документу до певного класу спостерігається кореляція - при збільшенні розміру навчальної вибірки збільшується і якість визначення ймовірності.