МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп’ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

ЧУМАДЕВСЬКА Христина Василівна

Метод кластеризації набору документів для ведення електронного документообігу / Method of Clustering a Set of Documents for Electronic Document Workflow

спеціальність: 122 - Комп'ютерні науки
освітньо-професійна програма - Комп’ютерні науки

Кваліфікаційна робота

Виконала студенка групи КНм-21
Х.В. Чумадевська
_____

Науковий керівник:
к.т.н., доцент Д.І. Загородня
_____

Кваліфікаційну роботу
допущено до захисту:
«___» _____ 20___ р.
Завідувач кафедри
_____ М.П. Комар

ТЕРНОПІЛЬ - 2023

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «магістр»
спеціальність: 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ М.П. Комар
«____» _____20__р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Чумадевська Христина Василівна
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи
Метод кластеризації набору документів для ведення електронного документообігу / Method of Clustering a Set of Documents for Electronic Document Workflow
керівник роботи к.т.н., доцент Д.І. Загородня
затверджені наказом по університету від 8 грудня 2022 року № 491.

2. Строк подання студентом закінченої кваліфікаційної роботи 1 грудня 2023 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити
- огляд предметної області;
- огляд літератури;
- аналіз вимог та постановка задачі дослідження;
- аналіз методів кластеризації;
- опис методу кластеризації модифікованим алгоритмом k-середніх;
- огляд методик оцінювання ефективност алгоритмів кластеризації
- опис засобів реалізації;
- реалізація методу кластеризації набору документів;
- Оцінювання ефективності методу кластеризації набору документів.

5. Перелік графічного матеріалу у роботі
- особливості електронного документообігу;
- візуалізація процесу кластеризації даних;
- розподіл наявних фільмів по п'ятьом кластерам.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 8 грудня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Аналіз предметної області та постановка задачі дослідження	12.2022 р. – 03.2023 р.	
2	Методи кластеризації для набору документів	03.2023 р. – 05.2023 р.	
3	Реалізація методу кластеризації набору документів	05.2023 р. – 11.2023 р.	
4	Повне завершення та представлення кваліфікаційної роботи на кафедру	01.12.2023 р.	

Студентка _____ Х.В. Чумадевська
підпис

Керівник роботи _____ к.т.н, доцент Д.І. Загородня
підпис

РЕЗЮМЕ

Кваліфікована робота на тему «Метод кластеризації набору документів для ведення електронного документообігу» на здобуття освітнього ступеня «Магістр» із спеціальності 122 «Комп'ютерні науки» освітньої програми «Комп'ютерні науки» написана обсягом в 80 сторінки і містить 13 ілюстрації, 2 таблицю, 3 додатки та 51 використаних джерел.

Метою кваліфікаційної роботи є визначення та дослідження методів кластеризації набору документів для організації і доступу до ефективного ведення електронного документообігу.

Методи дослідження: удосконалення вибраного алгоритму кластеризації, порівняння методу з використанням стандартних метрик ефективності та точності кластеризації для визначення переваги над існуючими методами.

Результати дослідження: вказують на потенціал методу кластеризації для оптимізації та організації документів у віртуальному середовищі. Вдосконалення методу набору документів для ведення електронного документообігу, що дало змогу уникнути основних недоліків вже існуючих методів і підвищити рівень ефективності кластеризації.

Результати роботи можуть успішно застосовуватися для підвищення, продуктивності та ефективності управляння документами через використання методу кластеризації в сфері електронного документообігу.

Ключові слова: ДОКУМЕНТООБІГ, МЕТОДИ КЛАСТЕРИЗАЦІЇ ДАНИХ, КЛАСТЕРИЗАЦІЯ, K-СЕРЕДНІХ, ПІДВИЩЕННЯ ТОЧНОСТІ КЛАСТЕРИЗАЦІЇ.

ABSTRACT

Qualification work on the topic " Method of Clustering a Set of Documents for Electronic Document Workflow " for the degree of "Master" in the field of 122 "Computer Science" of the educational program "Computer Science" is written with a volume of 80 pages and includes 13 illustrations, 2 tables, 3 appendices, and references to 51 sources.

The aim of the qualification work is to identify and investigate methods of clustering a set of documents to efficiently organize and access electronic document management.

Research methods include improving the selected clustering algorithm, comparing the method using standard metrics of clustering efficiency and accuracy to determine its advantages over existing methods.

Research results indicate the potential of the clustering method to optimize and organize documents in a virtual environment. The enhancement of the document set method for electronic document management has allowed overcoming the main drawbacks of existing methods and increasing the efficiency of clustering.

The results of the work can be successfully applied to enhance the productivity and efficiency of document management by using the clustering method in the field of electronic document management.

Keywords: DOCUMENT MANAGEMENT, DATA CLUSTERING METHODS, CLUSTERING, K-MEANS, CLUSTERING ACCURACY ENHANCEMENT.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

СЕД – Сервіс електронного документообігу

DVS – Document Management System

OPTICS – Ordering Points To Identify the Clustering Structure

BSCAN – Density-Based Spatial Clustering of Applications with Noise

VSM – Vector Space Model

ВСТУП

Актуальність теми. В сучасному цифровому світі обсяги електронних документів стрімко зростають, ставлячи перед організаціями завдання ефективного зберігання, організації та доступу до надмірної інформації. Електронний документообіг перетворюється на необхідну складову роботи будь-якої сучасної організації, проте його масштаби вимагають нові виклики щодо ефективного управління цими документами. У цьому контексті, метод кластеризації набору документів набуває особливого значення, пропонуючи інноваційні підходи до організації та каталогізації електронних матеріалів.

Інструменти узагальнення та візуалізації можуть допомогти користувачам зрозуміти інформацію, яка міститься у великій колекції документів. Сучасні методи візуалізації, доступні для цифрових бібліотек, складаються з двох основних компонентів: компонента агломерації, який ідентифікує кластери подібних документів, і компонента підсумовування, який представляє автоматично обчислений огляд документів у кожному кластері. Застосовуючи їх, користувачеві не потрібно переглядати всі документи в колекції один за одним, а замість цього можна отримати зображення високого рівня. Вважається, що кластеризація та класифікація є важливими методами семантичного аналізу в новому поколінні цифрових бібліотек (Додаток А).

Метод кластеризації документів стає невід'ємною частиною розв'язання цих проблем. Його застосування дозволяє автоматизувати процеси категоризації та організації документів на основі їх змісту та властивостей. Він забезпечує зручний пошук та доступ до інформації, спрощує управління документами та полегшує аналіз великих обсягів даних. Враховуючи ростуть потреби в ефективному управлінні даними і інформацією, метод кластеризації документів стає ключовим інструментом для сучасних організацій у контексті електронного документообігу.

Метод кластеризації набору документів для ведення електронного документообігу обґрунтовується потребою в управлінні електронними

документами у сучасному цифровому середовищі. Організації будь-якого типу та масштабу стикаються з великим обсягом інформації, яка надходить у вигляді електронних документів. Ця інформація потребує ефективного зберігання, обробки та доступу. Традиційні методи організації документів часто стають неефективними у забезпеченні швидкого пошуку, структурування та аналізу цієї інформації.

Цей метод актуальний у зв'язку з необхідністю вдосконалення систем управління документами, забезпеченням ефективного доступу до інформації та збільшенням продуктивності. Його застосування в контексті електронного документообігу є важливим кроком у напрямку оптимізації управління документами та підвищення ефективності роботи організацій у сучасному цифровому середовищі.

Мета і завдання дослідження. Метою кваліфікаційної роботи є дослідження методів кластеризації набору документів для ведення електронного документообігу.

Для досягнення поставлених цілей було визначено декілька завдань:

– дослідження актуальності методу кластеризації;

– аналіз відомих алгоритмів кластеризації;

– постановка задачі дослідження;

– удосконалення вибраного алгоритму;

– проведення експериментальних досліджень запропонованого методу кластеризації набору документів.

Об’єкт дослідження – процес кластеризації документів для ведення електронного документообігу.

Предмет дослідження – розробка та вдосконалення методу кластеризації, спрямованого на організацію та групування електронних документів у системах електронного документообігу.

Методи дослідження. Для вирішення поставленого завдання було використано наступні методи:

– аналіз літературних джерел;

– тестування алгоритмів кластеризації з метою оцінки їхньої ефективності та точності;

– удосконалення вибраного алгоритму кластеризації;

– порівняння методу з використанням стандартних метрик ефективності та точності кластеризації для визначення переваги над існуючими методами.

Наукова новизна одержаних результатів. Запропоновано метод кластеризації документів на основі модифікованого алгоритму k-середніх, який розбиває документи на кластери таким чином, щоб евклідова відстань між ними була мінімальною.

Практичне значення отриманих результатів. Досліджено алгоритм кластеризації k-середніх. Для кращого розрахунку було використані ефективні процедури кластеризації. Вивчення конкретного прикладу проводилося за допомогою метрики якості кластеризації: точності, запам’ятовування та f-міри. Всі дані підтверджуються позитивним результатом.

Публікації та апробація. Результати кваліфікаційної роботи були апробовані та опубліковані у матеріалах (Додаток Б):

– науково-практичної конференції молодих вчених і студентів «Інтелектуальні комп’ютерні системи та мережі», 2023 р.

– міжнародної науково-практичної інтернет-конференції «Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення», 2023р.

Кваліфікаційна робота складається із вступу, трьох розділів, висновків, списку використаних джерел та додатків.

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Електронний документообіг

В сучасному світі, перехід до електронного документообігу стає невід'ємною частиною стратегії більшості організацій та підприємств. Більшість країн, включаючи Україну, визнають важливість діджиталізації, що полягає в активному впровадженні цифрових технологій. Створення Міністерства цифрової трансформації в Україні у вересні 2019 року свідчить про стратегічний підхід до цього питання та прагнення ефективно використовувати потенціал цифрового розвитку.

Сучасні рішення у сфері цифровізації та автоматизації бек-офісних процесів дозволяють не тільки оптимізувати витрати, але й підтримують компанії в цифровій трансформації.

Впровадження системи документообігу дозволяє організувати бізнес-процеси та організувати їх за визначеними шляхами обігу для даного типу справи. Це полегшує точний моніторинг усіх кроків дій і звітування про запізнілі. Таким чином, організація може діагностувати місця, де слід вжити коригувальні дії. Швидший доступ до даних, які одночасно доступні кільком користувачам, полегшує прийняття важливих бізнес-рішень.

Електронний документообіг (також взаємозамінний документообіг або обіг електронних документів) – це процес передачі документів через різні етапи роботи в цифровій формі. У рамках цього процесу документи обробляються, передаються, зберігаються, перевіряються та підписуються в електронному вигляді та автоматично передаються на наступні етапи або користувачам.

Електронна обробка документів має на меті замінити традиційні паперові та ручні процеси, які є більш трудомісткими, дорогими та складними. Обробляючи документи в електронному вигляді, ви можете прискорити бізнес-процеси, підвищити ефективність роботи та знизити ризик втрати документів або

несанкціонованого доступу до конфіденційної інформації.

Програма управління електронними документами дозволяє обробляти через системи управління документами (DMS) або системи документообігу, які дозволяють відстежувати хід роботи, призначати завдання конкретним особам і автоматизувати документообіг між різними етапами роботи. Він також дозволяє використовувати такі інструменти, як шаблони документів, нагадування про крайні терміни та автоматично створені звіти.

Внутрішній документообіг включає створення, узгодження, підписання та використання документів всередині підприємства та подальше управління ними шляхом передачі в архів. Оцифрування таких документообігів, як правило, здійснюється за допомогою бухгалтерських програмних продуктів. Зовнішній електронний документообіг дає можливість обмінюватися юридично важливими документами (договорами, рахунками-фактурами, інвойсами, листами, запитами, звітами) між торговими партнерами та контролюючими органами. Реалізація таких документообігів можлива за допомогою спеціального програмного забезпечення або за допомогою онлайн-платформ для створення основних документообігів і звітів.

Кожен із цих менеджерів електронних документів має свої особливості та можливості (рисунок 1.1).

Серед основних особливостей внутрішнього електронного документообігу слід відзначити наступне:

– створити документи самостійно або використовувати шаблони, а також завантажувати документи в будь-якому форматі, створені за допомогою інших інформаційних систем;

– створення типів документів на основі інших типів документів з автоматичною передачею даних;

– перегляд створених або завантажених документів та підписувати їх кваліфікованим електронним підписом або іншим внутрішнім ідентифікатором;

– надсилання внутрішньої документації для перевірки або затвердження із зазначенням порядку уповноважених учасників маршруту та їх ролі в бізнес-

процесі;

– контроль процесу затвердження та підписання документів за допомогою індикаторів перегляду документів співробітників. Це дає можливість миттєво отримати інформацію про зміну статусу документа після виконання працівником дії та погодження;

Рисунок 1.1 – Особливості електронного документообігу

– зберігання документів в електронному форматі. Це дозволяє швидко отримати, перевірити (формат самого документа, виконавця і процедуру) і використовувати в подальшій роботі;

– зберігання документів на внутрішніх або зовнішніх серверах не тільки

економить значну частину ресурсів, доступних для друку, але також може використовуватися для резервування та обслуговування архівного простору;

– віддалене створення та/або використання документів за допомогою різноманітних технологічних пристроїв (комп'ютерів, смартфонів, планшетів) з віддаленим доступом з будь-якої точки світу 24/7.

Завдяки зазначеним вище функціям, запровадивши внутрішній електронний документообіг, можна скоротити час і витрати на реєстрацію, безпечне зберігання документів та зручний пошук, а також визначитися з технологією (порядком) передачі внутрішніх документів до компанії (від творця до особи, яка нею керує) також, що дуже важливо, керує базою даних віддалено.

Система керування документами – це система, яка надає такі послуги, як зберігання, контроль версій, метадані, безпека, індексування та функції пошуку. Велику кількість документів можна автоматично групувати в класи документів, які містять схожу інформацію. Для цього рекомендується використовувати методи кластеризації для групування документів.

Системи управління документами стають все більш важливими в управлінні документами. Елементами DMS документа є: інтегрує вилучення метаданих, збір, перевірку, індексування, зберігання, пошук, розповсюдження, безпеку, робочий процес, співпрацю, контроль версій, пошук, публікацію та відтворення. Крім того, деякі нові системи надають можливість пошуку та узагальнення тексту.

Програмне забезпечення машинного навчання надає базові методи інтелектуального аналізу даних шляхом вилучення інформації з необроблених даних, що містяться в базі даних. Процес зазвичай включає такі кроки:

- конвертувати дані у відповідний формат;
- очищення даних;
- висновки на основі отриманих даних.

Приклади СЕД:

– Google Drive;

– Dropbox;

- Box;
- Microsoft SharePoint;
- Microsoft OneDrive;
- Egnyte;
- eFileCabinet;
- DocuWare;
- Citrix ShareFile;
- M-Files;
- Google Cloud Search.

Інструменти та технології для електронного документообігу широко розповсюджені в організаціях різних сфер і діяльності.

1) Системи управління документами (DMS). Вони дозволяють зберігати, організовувати та керувати документами в електронному вигляді. Популярні DMS включають в себе SharePoint, Alfresco, і Documentum.

2) Електронний документообіг (EDM) і системи керування даними (DM). Вони автоматизують процеси обробки документів, включаючи їх розсилку, підписання, зберігання та архівування. Зокрема, можуть використовуватися такі системи, як M-Files, Laserfiche, та Hyland OnBase.

3) Електронний підпис і цифрова ідентифікація. Ці технології дозволяють забезпечити легальну валідність електронних документів і впроваджувати безпечний обмін даними. Наприклад, DocuSign або Adobe Sign.

4) Хмарні платформи для документообігу. Вони дозволяють зберігати документи в хмарних сховищах та використовувати їх для спільної роботи та доступу до даних. Наприклад, Google Workspace, Microsoft 365, Dropbox Business.

5) Бізнес-аналітика. Інструменти аналізу даних, такі як Tableau, Power BI або QlikView, можуть використовуватися для виявлення трендів, аналізу даних електронного документообігу та виведення корисної інформації для прийняття рішень.

6) Інтеграція з електронними системами управління бізнесом (ERP). Деякі рішення забезпечують можливість інтегрувати процеси управління документами з загальними системами управління організацією, такими як SAP, Oracle ERP Cloud, Microsoft Dynamics.

Ці інструменти та технології не лише полегшують управління електронними документами, але й покращують продуктивність, забезпечують безпеку та створюють ефективні механізми роботи з даними в сучасному бізнес-середовищі.

Джерела даних для методу кластеризації набору документів для ведення електронного документообігу включають в себе різні електронні документи, що зберігаються в системах електронного документообігу або в інших форматах, таких як PDF, Word, Excel, текстові файли тощо. Процес збору даних для кластеризації є наступними:

– збір електронних документів. Документи можуть бути зібрані з електронних систем управління документами (DMS), електронних поштових скриньок, локальних дисків або хмарних сховищ;

– фільтрація та очищення даних. Перш ніж застосовувати методи кластеризації, дані потрібно перевірити на наявність шуму, вилучити дублікати, а також виправити будь-які помилки або відсутні дані;

– векторизація текстової інформації. Для роботи з текстовими документами потрібно перетворити текст на числові вектори, наприклад, за допомогою методів Bag-of-Words або TF-IDF;

– визначення ознак та метаданих. Важливо визначити ознаки документів, такі як ключові слова, теми, дати створення, автори, категорії тощо, що допоможе побудувати модель кластеризації;

– трансформація та підготовка даних: Цей етап включає нормалізацію даних, кодування категоріальних ознак, масштабування тощо, для покращення роботи алгоритмів кластеризації.

– обробка даних. Перед застосуванням методів кластеризації варто зробити попередній аналіз та візуалізацію даних для зрозуміння їх структури та

особливостей.

Процес збору та підготовки даних визначає якість та ефективність подальшої кластеризації документів у системі електронного документообігу. Інструменти можуть бути недостатньо гнучкими чи потужними, щоб вважатися СЕД корпоративного рівня , але надають основні функції, які часто використовуємо в таких програмах.

Більшість організацій використовують СЕД. Поки є задіяні документи, зазвичай існує СЕД. Однак є кластери, які більш надійні щодо цих продуктів, ніж інші, наприклад:

– великі та середні компанії;
– державні відомства та установи;
– некомерційні організації та громадські організації.

Дійсно, невеликі підприємства, звичайні магазини та окремі працівники можуть обходитися без СЕД, але часто є певний спосіб організувати документацію.

Загалом, електронний документообіг має багато переваг перед паперовим і служив основним способом створення, передачі, передачі, надсилання та отримання ділових документів під час пандемії коронавірусу через обмеження щодо особистих та персональних документів. Необхідність дистанційного спілкування між торговими партнерами, співробітниками та іншими людьми. Впровадження як внутрішнього, так і зовнішнього електронного документообігу є одним із ключових напрямів підвищення ефективності, точності та якості інформаційного забезпечення управління. Кожен з них має свої особливості та функції. Тому компанії повинні бути зацікавлені в поєднанні обох типів в одному сервісі за оптимальних умов.

1.2 Опис задачі кластеризації

Кластеризація – це важливий процес інтелектуального аналізу текстів, який

використовується для отримання документів і отримання знань із документів на основі їхнього змісту. За останні роки використання документів в інтернеті значно зросло, і з покращенням якості та швидкості інтернету наше суспільство стало дуже залежним від інформації. Величезний обсяг даних, що генерується в процесі комунікації, є важливою інформацією, яка генерується щодня і зберігається у вигляді текстових документів, баз даних тощо.

Формально задачу кластеризації представляють наступним чином: нехай $X=\{x^1, x^2,\dots x^n\}$ – множина N об’єктів, заданих в n-вимірному векторному просторі ознак:

$$x^k = (x_1^k, x_2^k, \dots, x_n^k)^T, k = 1..N$$

Нехай Y = {1,2,3,…} буде набором номерів міток кластера та заданою функцією відстані між об’єктами. Найчастіше використовується евклідова метрика:

$$p_2(x, x') = \sqrt{\sum_{j=1}^{n} (x_j - x_j')^2}$$

Потрібно створити остаточну вибірку об'єкта X на K підмножин, що не перетинаються:

$$S_k, k = 1 \dots K;\ X = \cup_{k=1}^{K} S_k$$

Тому кожна підмножина (кластер) складається з об’єктів, близьких до метрики p, а об’єкти в різних кластерах істотно відрізняються. Алгоритм кластеризації – це функція $X \to Y$, яка відповідає будь-якому об’єкту $x \in X$ із номером кластера $y \in Y$.

Хоча множина Y часто відома заздалегідь, проте постає завдання визначити оптимальну кількість кластерів за тим чи іншим критерієм кластеризації.

В останні роки автоматизація збору інформації з багатьох типів систем пов'язана з великими обсягами даних. Як наслідок, було розроблено багато методологій, спрямованих на організацію та моделювання даних. Такі методології

широко використовуються в діагностиці, освіті страхуванні та багатьох інших сферах. Визначення, оцінка та застосування цих методологій є частиною галузі машинного навчання, яка стала основною підгалуззю підгалузі комп'ютерних наук і статистики через їхню вирішальну роль у сучасному світі.

Машинне навчання охоплює різні напрями, такі як регресійний аналіз , методи виділення ознак та класифікація - присвоєння класів об'єктам у наборі даних. Існує три основні підходи до класифікації: контрольована, напівконтрольована та неконтрольована класифікація [7]. Напівконтрольована класифікація відноситься до алгоритму, що використовує як мічені, так і немічені дані. Вони зазвичай використовуються коли ручне маркування набору даних стає дорогим. Неконтрольована класифікація також називають кластеризацією, полягає у визначенні класів з даних без наявності міток класів.

Кластеризація по тексту — це процес групування схожих документів разом на основі їхнього вмісту. Групуючи текст, ми можемо визначити закономірності та тенденції, які інакше було б важко помітити (рисунок 1.2).

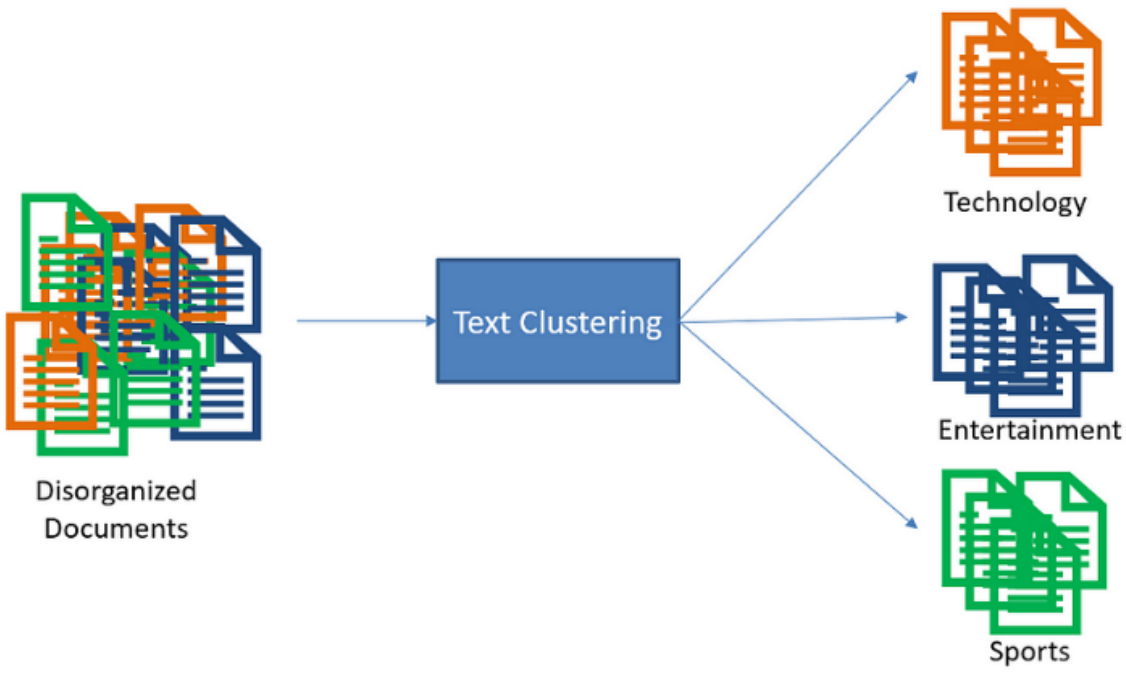

Рисунок 1.2 – Приклад кластеризації текстової інформації

Метою алгоритмів кластеризації є виявлення груп об'єктів або кластерів, всередині яких об'єкти більш схожі один на одного, ніж на об'єкти в інших кластерах. Метою алгоритмів кластеризації є виявлення груп об'єктів або кластерів, всередині яких об'єкти більш схожі один на одного, ніж на об'єкти в інших кластерах. Цей підхід до аналізу даних тісно пов'язаний із завданням створення моделі даних, тобто визначенням спрощеного набору властивостей, які можуть забезпечити інтуїтивне пояснення важливих аспектів набору даних. Методи кластеризації, як правило, є більш вимогливими, ніж контрольовані підходи, але забезпечують краще розуміння складних даних.

У сучасному світі інформаційних технологій широко застосовуються методи кластеризації. Зокрема, її використовують для виявлення фейкових новин на основі їхнього змісту. З її допомогою можна визначити, чиє інформація достовірною, чи ні, ефективно фільтрувати спам у вхідних електронних листах, аналізувати групи в маркетингу та продажах електронної пошти, а також для аналізу цільових груп у маркетингу та продажах.

Алгоритми кластеризації можуть об'єднувати людей в групи, наприклад, людей зі схожими рисами особистості, чи тих що здійснювали певні покупки . Це дозволяє компаніям максимізувати віддачу від маркетингових інвестицій.

Також слід також відзначити важливість кластеризації в процесі аналізу документів. Це корисно для організації та класифікації документів. Корисно при класифікації документів відповідно до текстового вмісту в документі. У цьому випадку кластеризація є корисним інструментом. У деяких реалізаціях її можна використовувати як алгоритм кластеризації. Цей алгоритм здатний розпізнавати тексти і групувати їх за різними темами. Ця методика дозволяє впорядковувати схожі документи набагато швидше.

За допомогою кластеризації в медицині при МРТ-скануваннях розпізнають злоякісні ураження і диференціюють різні типи тканин для різних цілей. Також кластерний аналіз може бути використаний для аналізу шаблонів резистентності до антибіотиків, класифікувати антибіотики відповідно до їхньої антимікробної активності.

В інформаційних технологіях кластеризація особливо корисна при розробці програмного забезпечення, тому що вона допомагає зменшити несуттєві залежності в коді шляхом реструктуризації. Кластеринг може бути використаний для визначення контурів або для поділу на різні області (Рисунок 1.3).

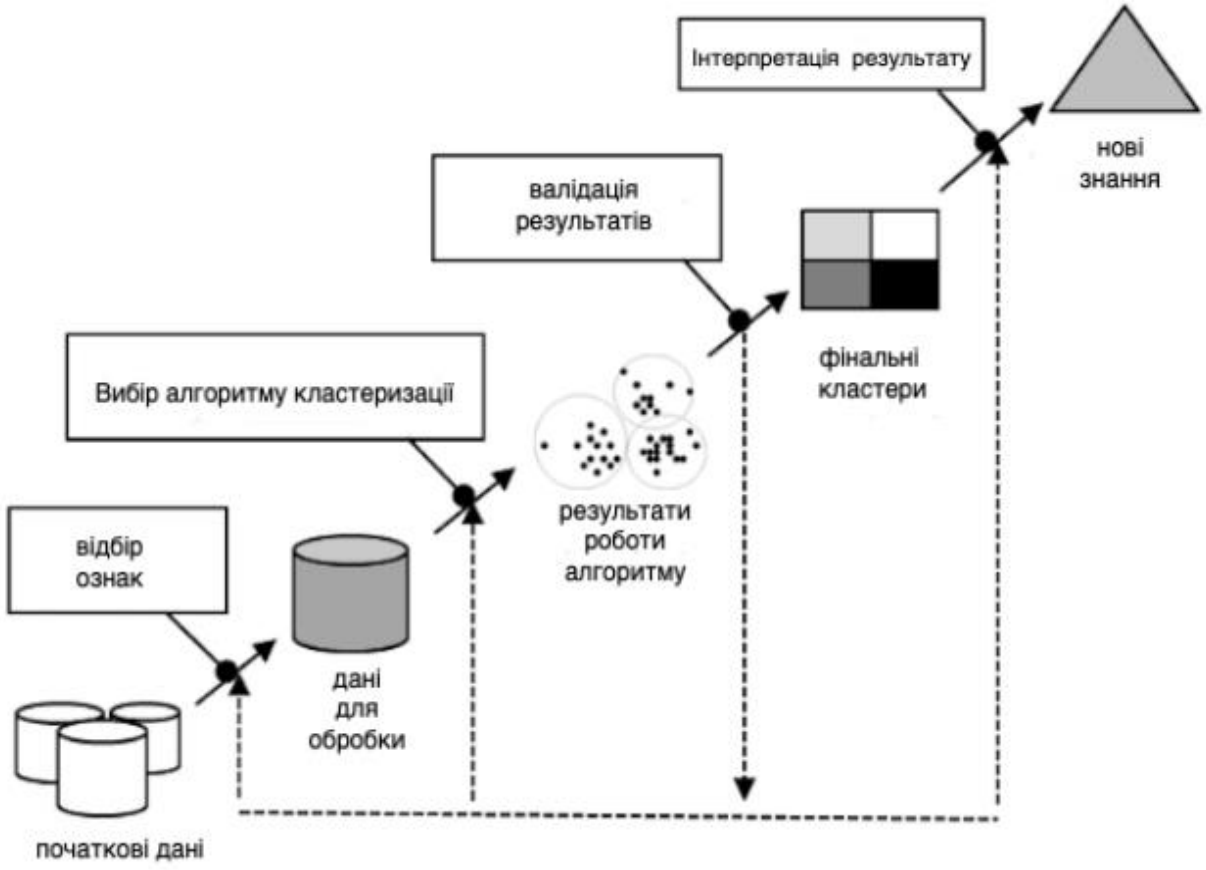

Рисунок 1.3 - Візуалізація процесу кластеризації даних

Наведемо приклади сфер застосування кластеризації тексту. Кластеризацію використовують наступні дисципліни та галузі:

- інформаційний пошук – методи кластеризації використовувалися в різних програмах для пошуку інформації, включаючи кластеризацію великих наборів даних. У пошукових системах кластеризація тексту відіграє вирішальну роль у покращенні ефективності пошуку документів шляхом групування та індексування пов’язаних документів.
- інтернет речей – зі швидким розвитком технологій, збір даних в

Інтернеті речей передбачає використання глобальної системи позиціонування, технології радіочастотної ідентифікації, датчиків і різних інших пристроїв Інтернету речей. Методи кластеризації використовуються для розподіленої кластеризації, яка є важливою для бездротових сенсорних мереж.

- біологія – коли кластеризувати гени та зразки в експресії генів, характеристики даних експресії генів стають значущими. Вони можуть бути класифіковані на кластери на основі їх моделей експресії.
- промисловість – підприємства збирають великі обсяги інформації про поточних і потенційних клієнтів. Для подальшого аналізу клієнтів можна розділити на малі групи.
- клімат – розпізнавання глобальних кліматичних моделей вимагає виявлення моделей в океанах і атмосфері. Кластеризація даних спрямована на виявлення моделей атмосферного тиску, які суттєво впливають на клімат.
- медицина – кластерний аналіз використовується для диференціації підкатегорій захворювань. Він також може виявити моделі захворювання в часовому або просторовому розподілі.

1.3 Огляд літератури

Еверітт [1] визначив кластер як «набір сутностей, які схожі, і сутності з різних кластерів не однакові». Прикладом раннього дослідження кластеризації в інформаційній науці є робота Джардіна та ван Рейсбергена [2] . Ідея кластеризації полягала в тому, що якщо певні документи відповідають запиту користувача, документи в тому самому кластері також, швидше за все, будуть релевантними.

Огляд використання програм кластеризації для пошуку інформації зробив Расмуссен [3]. Вона визначила наступні основні проблеми із застосуванням різних методів кластеризації в області аналізу тексту:

1) Складність оцінки достовірності отриманих результатів.
2) Вибір відповідних атрибутів для кластеризації.

3) Вибір відповідного методу кластеризації.
4) Висока вартість обчислювальних ресурсів.

Останнім часом методи візуалізації інформації відродили інтерес до кластеризації. Ідея багатьох із цих методів, які дозволяють візуалізувати великі колекції документів, полягає в об’єднанні подібних документів у кластери та представленні високорівневого підсумку кожного кластера. Таким чином, користувачеві не потрібно переглядати схожі документи або цілі документи, щоб ознайомитися з колекцією. Це значно зменшує надмірність і когнітивний попит.

Завдання кластеризації тексту нагадує завдання категоризації тексту, яке також інтенсивно досліджувалося вченими з інформації. За визначенням, категоризація тексту – це віднесення текстів природною мовою до однієї або кількох попередньо визначених категорій на основі їхнього змісту.

Однак завдання кластеризації є більш складним, оскільки не існує попереднього набору категорій, створених експертами-людьми. Наслідком машинного навчання є те, що на відміну від контрольованих методів, які використовуються для категоризації, для кластеризації можна використовувати лише неконтрольовані методи. Завдання кластеризації має більше ступенів свободи: не лише призначення рішень, але й рішення про те, скільки кластерів створити та які типи документів призначити кожному кластеру.

Традиційно в інформатиці методи кластеризації оцінювалися разом із завданнями пошуку. Наприклад, Херст і Педерсен [4] оцінили точність кластеризації на основі частки відповідних документів, знайдених у найбільшому кластері. Було виявлено напрочуд мало досліджень, які включають методологічну оцінку методів кластеризації на основі подібності між отриманими розділами та кластерами, створеними експертами-людьми.

Сахамі та ін. [5] базували свої вимірювання на тому, чи була пара об’єктів віднесена до одного класу людьми-експертами та системою. Для людських експертних оцінок вони використовували створені вручну категорії. Замір та ін.[6] використали подібне вимірювання для тестування кластеризації, застосованої до колекції, створеної шляхом злиття кількох менших колекцій веб-документів на

різні теми.

У попередніх дослідженнях кластеризації документів контрольні колекції створювалися шляхом об’єднання документів на різні теми. Як правило, більшість моделей головним чином зосереджені на навчанні репрезентації на основі локальних співуживань слів. Розуміння того, як працює модель, має вирішальне значення для використання та розробки методів кластеризації тексту. Кластеризація тексту, зазвичай, містить різні кроки, які можна застосувати, як показано на рисунку 1.4.

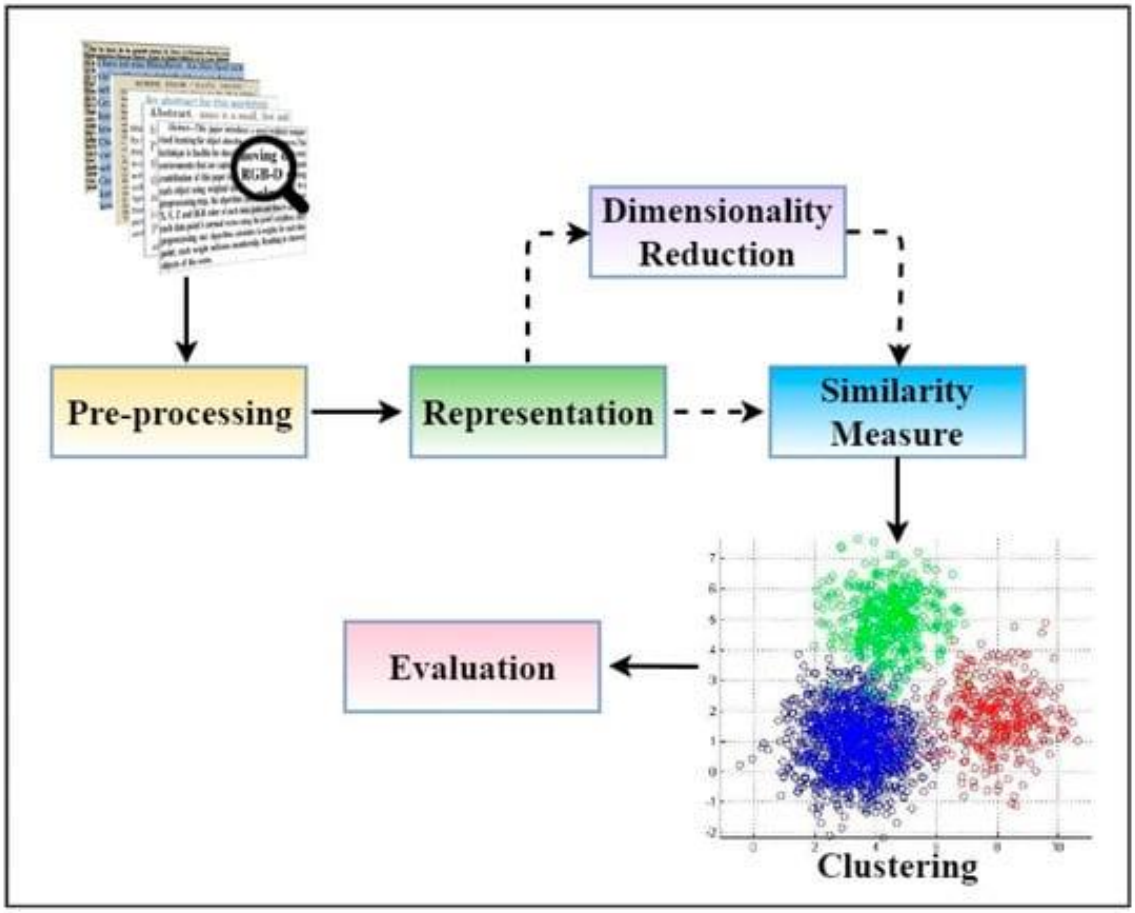

Рисунок 1.4 – Компоненти для кластеризації текстових даних

Попередня обробка – це перший крок у кластеризації тексту. Дані необхідно очистити, видаливши непотрібні символи, слова та цифри. Потім можна застосувати методи подання тексту. Попередня обробка відіграє важливу роль у створенні ефективної системи кластеризації, оскільки короткі текстові дані

(оригінальний текст) непридатні для використання безпосередньо для кластеризації.

Представлення – документи та тексти є наборами неструктурованих даних. Ці неструктуровані дані потрібно перетворити на структурований простір ознак, щоб використовувати математичне моделювання під час кластеризації. Стандартні методи представлення тексту можна розділити на методи, засновані на представленні корпусу, і на основі зовнішнього знання.

Зменшення розмірності – тексти або документи, часто після представлення традиційними техніками, стають багатовимірними. Процедури кластеризації даних можуть уповільнюватися через великий час обробки та складність зберігання. Зменшення розмірності є стандартним методом вирішення такого роду проблем. Багато науковців використовують зменшення розмірності, щоб зменшити час застосування та пам’ять, а не ризикувати падінням продуктивності. Зменшення розмірності може бути ефективнішим, ніж розробка недорогого представлення.

Міра подібності – це фундаментальна сутність в алгоритмі кластеризації. Вона полегшує вимірювання подібних об’єктів, групування найбільш подібних об’єктів і елементів і визначення найкоротшої відстані між пов’язаними об’єктами. Іншими словами, відстань і подібність мають обернену залежність, тому вони використовуються як взаємозамінні. Векторне представлення елементів даних зазвичай використовується для обчислення показників подібності/відстані.

Методи кластеризації – є вирішальною частиною будь-якої системи кластеризації тексту. Не можна вибрати найкращу модель для системи текстової кластеризації без глибокого концептуального розуміння кожного підходу. Метою алгоритмів кластеризації є створення внутрішньо узгоджених кластерів, які явно відрізняються один від одного.

Оцінка – це останній крок кластеризації тексту. Перед застосуванням або створенням техніки кластеризації тексту необхідно зрозуміти, як працює модель. Для оцінки кластеризації доступно кілька моделей.

1.4 Аналіз вимог та постановка задачі дослідження

Оцінка якості результатів кластеризації (і, отже, ефективності всієї процедури) є суб'єктивним процесом і залежить насамперед від вимог, які висуваються до процедури кластеризації. Ці вимоги часто залежать від фактичної області застосування кластеризації. Тому необхідно визначити вимоги до методів кластеризації тексту, спеціально розроблених у рамках цього дослідження.

Основною особливістю процедури кластеризації, спільною для всіх сфер практики, є вимога до інтерпретабельності результатів, отриманих наприкінці процедури. Результати кластеризації зазвичай потребують розуміння та інтерпретації користувачами методу.

Вимога інтерпретованості також породжує іншу вимогу: ексклюзивність кластеризації. Це означає, що між документами і кластером існує взаємозв'язок один-до-одного. Тобто текст може належати лише одному кластеру. Це пояснюється тим, що більшість застосувань методів кластеризації стосуються чіткого присвоєння того чи іншого тексту відповідному кластеру.

Окрім вимоги інтерпретабельності, необхідність практичного застосування методу та практична застосовність алгоритму призводять до появи кількох інших вимог до методу кластеризації.

Досвід роботи з алгоритмами обробки тексту показує, що для алгоритмів верхня межа прийнятної швидкості роботи є квадратичною залежністю від кількості тексті. Тому через таку залежність від часу роботи алгоритм не погіршує свою продуктивність із збільшенням кількості документів у корпусі.

Необхідною умовою для методів кластеризації є наявність синонімів у тексті та методу розв'язання синонімів. Це пов'язано з тим, що, на відміну від людей, самі комп'ютери не мають здатності швидко визначати контекстне значення кожного терміна, і саме тому при розробці математичних методів обробки текстів є однією з головних труднощів.

У більшості практичних застосувань методів кластеризації часто існують припущення щодо кількості кластерів у корпусі документів на початку

кластеризації. Тому бажано мати можливість вручну встановити кількість результуючих кластерів для методу. При цьому також необхідно залишити відкритою можливість автоматичного визначення кількості кластерів.

Іншою вимогою до методів кластеризації є стабільність, оскільки текстовий корпус розширюється до більших підмножин загального набору, до якого він належить. Ця вимога може бути визначена як статистичні властивості результатів кластеризації. Ця властивість методів кластеризації особливо важлива при розгляді динаміки розподілу документів за кластерами в часі.

Якщо користувач, який використовує техніку кластеризації, насправді зацікавлений лише в результатах кластеризації для фіксованого набору текстів, які вважаються немасштабованими, тоді результати можна розглядати, властивості масового тексту. І навпаки, якщо задача розглядає досліджуваний корпус лише як частину генеральної множини, то з методу кластеризації природно вимагати, щоб отримані результати характеризували загальну множину в цілому, так що кластери отримані з різних вибірок цей загальний набір, а подібні.

Отже, найважливішими властивостями, якими повинен володіти метод кластеризації, щоб бути практичним у контексті цього дослідження, є:

– інтерпретація знайдених кластерів;

– зв’язки між документами та кластерами;

– час роботи алгоритму кластеризації повинен бути обмежений квадратичною залежністю від кількості документів у корпусі;

– існування синонімів і засобів для знаходження синонімів у методах кластеризації;

– наявність можливості ручного призначення кількості кластерів у методах кластеризації;

– статистична значущість групування тексту в кластери.

Висновки до розділу 1

1. Проведено аналіз предметної області, основних елементів та аспектів внутрішнього та зовнішнього електронного документообігу. Проаналізовано системи керування документами та інструменти підтримки ефективності даного процесу. Це дало змогу виділити основні елементи та інструменти необхідні для електронного документообігу.

2. Здійснено опис задачі кластеризації, проаналізовано сфери застосування процесу кластеризації, що дало змогу визначити основні критерії для визначення кластерів у відповідних областях даних.

3. Проведено огляд літератури та основних елементів процесу кластеризації, що дало змогу сформувати основні напрямки дослідження та підтвердити актуальність теми.

4. Проаналізовано вимоги до процедури кластеризації, що дало можливість сформувати вимоги до розроблюваного методу кластеризації документів.

РОЗДІЛ 2 МЕТОДИ КЛАСТЕРИЗАЦІЇ ДЛЯ НАБОРУ ДОКУМЕНТІВ

2.1 Методи кластеризації

Кластеризація передбачає групування набору подібних точок даних разом, які таким чином утворюють кластер. Кластер вважається хорошим, якщо подібність між кластерами (точки даних у межах одного кластера) висока, а подібність між кластерами (точки даних за межами кластера) низька. Кластеризацію також можна розглядати як техніку стиснення даних, у якій точки даних кластера можна розглядати як групу. Кластеризація також називається сегментацією даних , оскільки вона розбиває дані таким чином, що група подібних точок даних утворює кластер (рисунок 2.1).

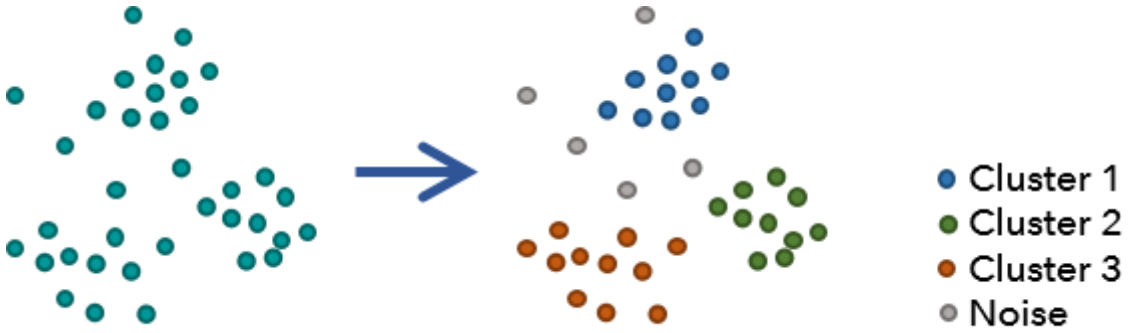

Рисунок 2.1 – Приклад кластеризаціїї

Найпоширеніші методи кластеризації:

- методи розбиття;
- ієрархічні методи;
- методи на основі щільності;
- модельні методи.

K-Means Clustering — це класичний підхід до кластеризації. K-Means ітеративно переміщує центри кластерів, обчислюючи середнє значення кластера. Спочатку K-Means вибирає k центрів кластерів випадковим чином. Відстань обчислюється між кожною точкою даних і центрами кластерів (зазвичай

використовується евклідова відстань). Точка даних призначається кластеру, до якого вона дуже близька. Після того, як усі точки даних присвоєно кластеру, алгоритм обчислює середнє значення точок даних кластера та переміщує центри кластера до відповідного середнього значення кластера. Цей процес триває до тих пір, поки центри кластерів не зміняться (див.рисунок 2.2).

Рисунок 2.2 – Робота алгоритму k-середніх

Перевагою використання k-середніх є масштабованість. Алгоритм добре працює з величезними даними. Основним недоліком використання k-середніх є його чутливість до викидів. Викиди сильно впливають на обчислювальні засоби кластера. Результати кластерів змінюються залежно від значення k і початкового вибору центрів кластерів. Алгоритм k-середніх добре працює лише для сферичних даних і погано працює з довільними формами даних.

K-Means добре працює для безперервних даних. Алгоритм дуже схожий на K-Means, але замість обчислення середнього значення кластера K-Modes обчислює моду (значення, яке часто зустрічається) кластера. Спочатку K-Mode вибирає k центрів кластерів випадковим чином. Подібність обчислюється між кожною точкою даних і центрами кластерів. Точка даних призначається кластеру, до якого вона має високу схожість. Після того, як усі точки даних призначено

кластеру, алгоритм обчислює режим точок даних кластера та переміщує центри кластера у відповідний режим кластера. Цей процес триває до тих пір, поки центри кластерів не зміняться.

Алгоритм ієрархічної кластеризації створює ієрархію або дерево з об'єктів даних. Цей метод може початися з кластера, а потім розділити його на менші кластери (цей метод називається Divisible або Topdown), або почати з об'єкта, а потім згрупувати пов'язані кластери разом (метод відповідно Agglomerative або Bottom-Up).

Починаючи з кожного з n об'єктів у кластері, процедура сукупної ієрархічної кластеризації групує два найближчі (або найбільш схожі) об'єкти в кластер, зменшуючи кількість кластерів до n- перших. Процес повторюється, доки всі об'єкти не будуть згруповані в кластер із n об'єктів. Кластер з використанням параметрів за замовчуванням, наданих програмним забезпеченням. Тому необхідні експерименти для оцінки та порівняння ефективності алгоритмів кластеризації в сценаріях оптимізації та за замовчуванням.

Цей метод має більше обмежень, ніж інші типи кластеризації, але він ідеальний для певних типів наборів даних. На рисунку 2.3 відображенні загальні принципи роботи ієрархічних висхідних і низхідних алгоритмів.

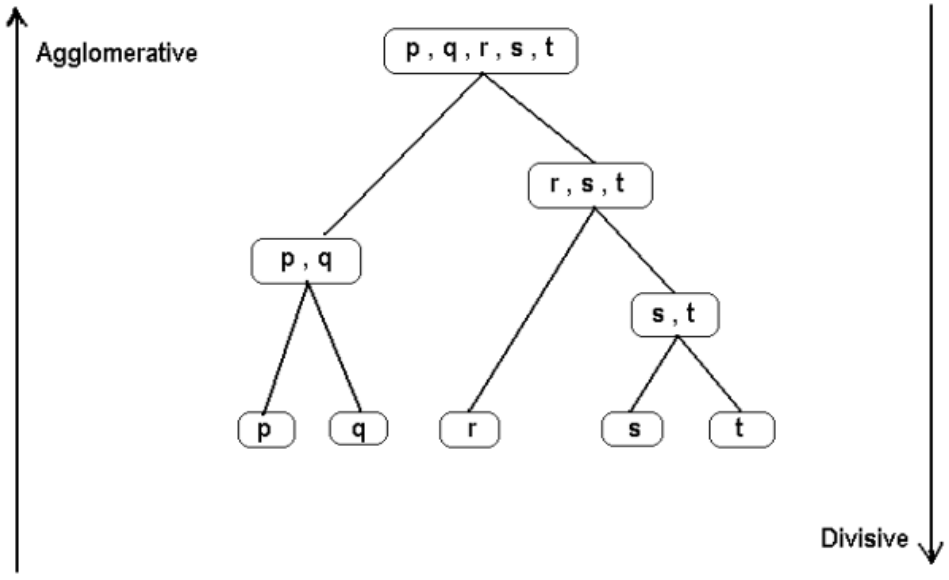

Рисунок 2.3 – Приклад ієрархічної кластеризації

Обмежене використання агломеративних алгоритмів кластеризації пояснюється тим, що будь-яке розділення або злиття є остаточним після завершення і не може бути скасовано, що може спричинити проблеми з даними із за створенням шуму. Знаменитий алгоритм BIRCH (збалансоване ітераційне скорочення та кластеризація з використанням ієрархії) намагається вирішити цю проблему, але його недоліком є те, що його важко масштабувати через обчислювальну складність і масштабованість також погано показує шум, що важливо для обробки великих даних Крім того, підходи «зверху вниз» і «знизу вверх» вимагають ієрархічної структури у вхідних даних.

Методи на основі щільності поділяється на DBSCAN та OPTICS. DBSCAN (Density-Based Spatial Clustering of Applications with Noise, рис 2.4) — це алгоритм кластеризації на основі щільності, який добре працює з даними довільної форми. DBSCAN знаходить щільні точки даних і призначає їх кластеру, залишаючи менш щільні точки даних окремо від кластера.

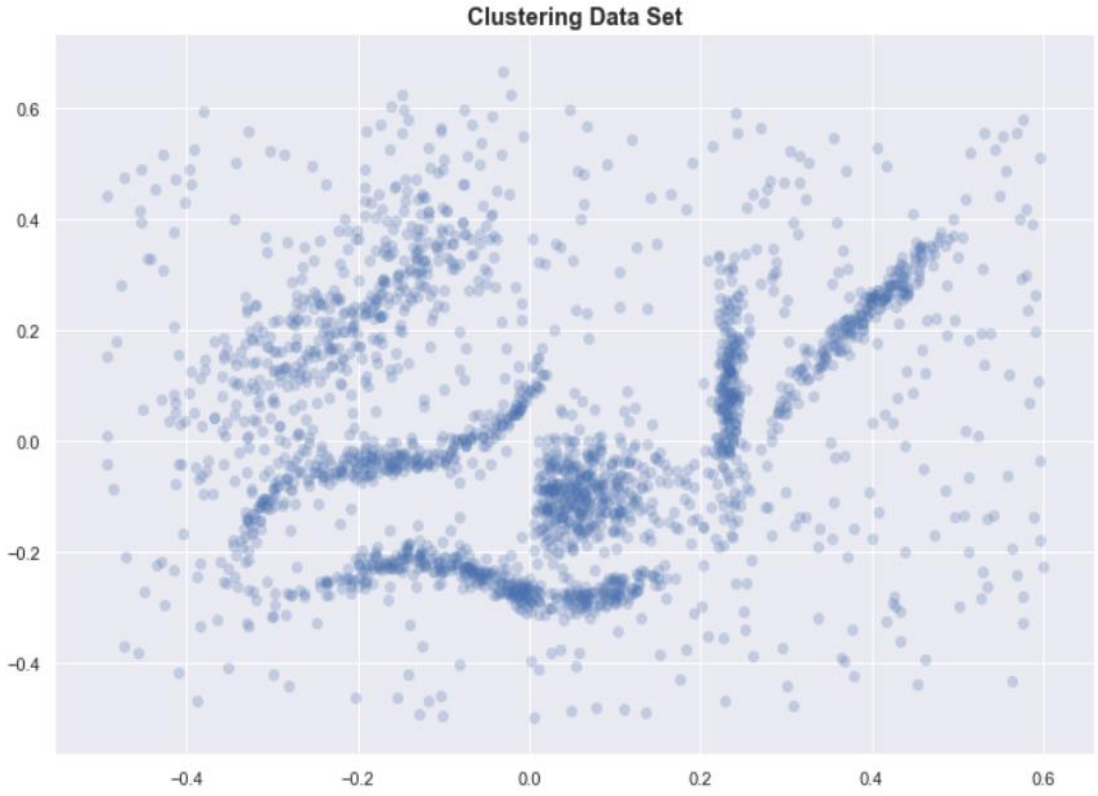

Рисунок 2.4 – Візуалізація датасету різної щільності

Термінології DBSCAN:

- якщо точка даних q знаходиться в межах радіуса є іншої точки даних p , тоді точка даних q вважається є-околицею (епсилон-околицею) точки даних p;
- точка даних p називається основним об’єктом, якщо є-околиця точки p складається зі значення, зазначеного в MinPts (мінімальні точки). Наприклад, розглянемо MinPts = 5, кажуть, що p є основним об’єктом, якщо є-околиця p складається з 5 точок даних;
- точка даних p називається прямодосяжною за щільністю з точки q , якщо точка p знаходиться в межах є-околиці q.

DBSCAN перевіряє є-околиці кожної точки даних. Якщо p є основним об’єктом і його точки даних у є-околиці вищі за значення MinPts, він утворює новий кластер навколо основного об’єкта p . DBSCAN ітераційно знаходить точки даних, доступні безпосередньо за щільністю, і може об’єднати кілька інших кластерів. Процес триває, доки не залишиться точок для аналізу.

Недоліком використання DBSCAN є наявність гіперпараметрів є та MinPts. Отримані результати змінюються залежно від значень, вибраних для гіперпараметрів.

OPTICS (Ordering Points To Identify the Clustering Structure) — це алгоритм кластеризації, який усуває недоліки, з якими стикається використання гіперпараметрів у DBSCAN. OPTICS дуже схожий на DBSCAN, але використовує гнучке значення для радіуса є.

Термінології OPTICS:

- найменше значення для ϵ' (менше значення для ϵ'), яке робить точку p основним об’єктом, називається базовою відстанню p;
- максимум основної відстані p і евклідової відстані між p і q називається відстанню досяжності точки q відносно іншого об’єкта p.

Проте методи кластеризації, незважаючи на свою потужність, стикаються з кількома проблемами:

- вибір оптимальної кількості кластерів. Визначення кількості кластерів може бути складною задачею, особливо коли немає чіткої попередньої інформації

про кількість груп у даних;

− чутливість до початкових параметрів. Деякі методи, зокрема k-середніх, можуть бути чутливими до початкового вибору центрів кластерів. Це може призвести до різних результатів при різних запусках алгоритму;

− робота з різнорідними та великими даними. Виклик становить обробка великих обсягів даних або даних з великою кількістю атрибутів. Деякі методи можуть виявитися неефективними або непрактичними для обробки таких обсягів;

− інтерпретація результатів. Отримані кластери можуть бути складно інтерпретовані. Іноді потрібно здійснювати додатковий аналіз для зрозуміння сутності та значимості кожного кластера;

− відсутність врахування контексту. Деякі методи можуть виявитися менш ефективними у випадках, коли важливий контекст чи взаємозв'язки між даними, наприклад, у випадку текстової інформації;

− обробка великого обсягу категоріальних даних. Для методів кластеризації, які використовують велику кількість категоріальних даних, можуть виникати проблеми з обробкою цієї інформації.

Розуміння цих викликів є ключовим для розвитку більш точних, ефективних та універсальних методів кластеризації даних.

Також важливим аспектом є кластеризація тексту. Через природу текстових даних і складність природної мови є складним завданням кластеризації. Деякі з основних проблем у кластеризації тексту включають:

1) Висока розмірність: текстові дані часто представляють як розріджену матрицю великої розмірності, що ускладнює використання традиційних алгоритмів кластеризації.

2) Шум і викиди: текстові дані можуть мати шуми, як-от орфографічні помилки, опечатки та нерелевантну інформацію, через що важко знайти шаблони, які щось означають.

3) Категоричні дані : текстові дані часто є категоричними, що означає, що вони не мають природного відчуття відстані чи подібності.

4) Робота з синонімами та полісемією: слова в текстових даних часто мають більше ніж одне значення, що ускладнює визначення справжнього значення.

5) Обробка розріджених даних: текстові дані часто розріджені, тобто багато слів не відображаються в більшості документів. Це ускладнює пошук шаблонів у даних, які щось означають.

6) Робота з великими наборами даних: кластеризація великих наборів даних може бути дорогою з точки зору обчислень і вимагати великого обсягу пам’яті.

7) Обробка нових слів : Алгоритми кластеризації навчаються на фіксованому наборі даних, тому вони можуть не мати змоги обробляти нові слова в навчальному наборі даних.

8) Масштабованість : алгоритми кластеризації повинні добре масштабуватися зі збільшенням розміру даних; інакше вони можуть стати непрактичними для використання з великими наборами даних.

9) Оцінка : результати кластеризації важко оцінити, оскільки немає єдиної правильної відповіді щодо кластеризації, а метрика оцінки залежить від програми та набору даних.

Незважаючи на ці проблеми, кластеризація тексту все ще є цінною технікою для отримання інформації з текстових даних. Як результат, його можна використовувати в широкому діапазоні застосувань. Дослідники винаходять нові методи та алгоритми вирішення цих проблем. Отже, нові методи на основі глибокого навчання можуть впоратися зі складністю текстових даних.

2.2 Опис методу кластеризації модифікованим алгоритмом k-середніх

Процес кластеризації документів складається з кількох етапів. Набір даних попередньо обробляється, щоб створити модель векторного простору (VSM) із набором маркерів. VSM – це процедура пошуку моделей TF-IDF. Міра подібності використовується для обчислення відстані між різними кластерами.

Модифікована кластеризація k-середніх – це алгоритм на основі розділів.

Це метод, який використовується для ініціалізації центроїда. Було запропоновано кілька інших алгоритмів для кластеризації документів, включаючи метод навпіл k-середніх, який ділить набір усіх точок на два кластери, вибирає один із них, ділить і повторює процес, доки не буде створено k кластерів. Гібридна кластеризація бісектриси k-середніх поєднує бісектрису k-середніх та алгоритми ієрархічного розбиття для отримання оптимальної кластеризації. Новий алгоритм використовується для автоматичної кластеризації та усуває недоліки алгоритму k-середніх.

Використовуються дерева подібності документів для кластеризації документів, щоб розділити послідовності фраз і слів у документах. Для кластеризації фрагментів на основі подібності використовується метод перехресного проходу з k-середніми.

Алгоритм k-середніх добре працює для малих наборів документів, проте зі збільшенням їх кількості аогоритм може не показати ажаних результатів. Результат згенерованих кластерів залежить від початкового значення випадково вибраних центроїдів, і він працює лише для глобальних кластерів. Метод k-середніх базується лише на відборі попередньо визначених кластерів. Щоб вирішити ці проблеми, було вирішено розпочати кластеризацію документів за допомогою модифікованого алгоритму k-середніх для покращення значення F-міри.

Попередня обробка виконується над звичайними текстовими документами і генерує набір токенів для виводу в Vector Space Model (VSM). Ця методика забезпечує оптимальну якість кластерів. Основні етапи препроцесорної обробки полягають у наступному:

1. Фільтрація: для видалення розділових знаків і спеціальних символів.
2. Токенізація: для розбиття токенів на окремі слова та токени.
3. Контроль видалення слів: слова, що не мають значення, видаляються.
4. Стеммінг: утворюється основна форма слів.
5. Обрізка: для видалення низькочастотних слів.

Пошук термінів знаходить ексклюзивні терміни з кожної доступної категорії. Запропонована робота присвоює кожному терміну порогове значення як вагу. Якщо частота терміна більша за порогове значення, то значення додається, інакше відхиляється.

Для всіх слів у наборах термінів

If (tf (i) > threshold)
Add to set
End

Вилучення функцій використовується для видалення набору ключових слів з документів. VSM – це техніка пошуку в інтелектуальному аналізі даних, також відома як модель "частота терміна - частота документа", тобто модель TF-IDF (Term Frequency Inverse Document Frequency). Це стандартна алгебраїчна модель представлення тексту. Кожен документ представляється у вигляді n-вимірного вектора за допомогою вектора ознак. Значення кожного елемента вектора відображає важливість відповідної ознаки в документі. За допомогою цієї моделі схожість між документами може бути вимірюється шляхом обчислення відстані між векторами документів. Якщо документи містять однакові ключові слова, вони вважаються схожими. Частота терміна tf (i, j) – це кількість разів, коли термін і зустрічається в документі. Порівняно з tf та булевою схемою відбору ознак, результати показують, що tf-idf найкраще підходить для створення кластерів. Частота терміна нормалізується відносно максимальної частоти всіх термінів, що зустрічаються в документі:

$$Tf\ (i, j) = freq\ (i, j) / (max\ \{f(x, j): w€j\})$$

де *i* – термін у документі *j*, а x – будь-який член із максимальною часткою.

Подібним чином частота терміну в документі – це кількість документів, яких зустрічається термін *i*. Розраховується як:

$$Idf\ (i, j) = log\ (D/dfi)$$

Евклідова схожість - це найпоширеніша міра схожості для обчислення схожості між двома документами. Обчислюється як:

$$S=S+ ((a\ (t)-b\ (k))^{*1/2})$$

Оцінка ефективності кластеризації вимірюється за допомогою F-міри. F-міра використовується для порівняння схожості двох кластерів. Це комбінація точності (P) та міри пригадування (Recall). Вона визначається за формулою:

$$F\text{-міра}= (2*PR)/ (P+R)$$

Точність (P) визначається як відношення кількості позитивних результатів (T_P) до кількості позитивних результатів плюс кількість хибних результатів (F_p)

$$P=(T_P)/(T_P + F_P)$$

Відгук (R) визначається як відношення кількості істинно позитивних спрацьовувань (T_P) до кількості істинно позитивних спрацьовувань плюс кількість помилково негативних результатів (F_N)

$$R=(T_P) / (T_P + F_N).$$

Наступний крок – це покращений метод k-середнє. Нехай задано вектор документів [x1 ,x2 ,. .xn]. Покращений алгоритм кластеризації k-середніх розбиває n документів на k кластерів таким чином, щоб евклідова відстань між ними була мінімальною. Спочатку він пророкує центр вручну, а потім виконує k-середніх на наборах даних.

Робота виконується на наборі даних за наступним алгоритмом представлений в псевдокоді:

Input: X={x1,x2….xi….xn}// набір із n точок даних

K // кількість клестерів

Output: Набір із K кластерів.

Потрібно визначити початкові центроїди кластерів за допомогою алгоритму. Віднести кожну точку даних до найближчого кластера. Алгоритм працює у два етапи. На першому етапі обчислюються початкові центроїди для підвищення точності, а на другому – точки даних розподіляються по кластерах шляхом обчислення евклідової відстані між ними.

Для початкового центрального визначення потрібні такі вхідні дані: VSM, K, n, terms, w. Вихідні дані: centers.

Наступним потрібно створити початкову центральну матрицю для кластерів при тому встановивши за замовчуванням нуль.

For Ki=1 to K

center (Ki: T: T+N-1) =W

T =T + N

End

Розподіл точок даних за кластерами

Input: X={x1,x2.....xi .....xn}

C={c1,c2.....ck}

Output: set of clusters.

Наступні кроки:

1. Обчислити евклідову відстань для кожної точки даних в X до всіх центроїдів.
2. Для кожної точки даних в X знайти найближчий центроїд і визначити кластер.
3. Для кожної точки даних xi, обчислити його відстань від центроїда до поточного найближчого кластера (представлений в псевдокоді):

If

Distance <= present nearest distance

Then

Cluster remain same

Set clusteredID (i) =j.

Set near_dist=d (Xi, cj).

Else

For every centroid, compute the distance and assign cluster to nearest centroid.

Set clusteredID (i) =j.

Set near_dist=d (Xi, cj).

End

End

Перераховуються центроїди поки результати не залишаються не змінними.

2.3 Методики оцінювання ефективності алгоритмів кластеризації

Оцінка ефективності алгоритмів кластеризації включає в себе кілька підходів і методів для оцінки якості отриманих кластерів [9]. Таких як:

1) Внутрішні методи.
2) Зовнішні методи.
3) Відносні методи.
4) Методи стабільності.
5) Візуальні методи.
6) Гібридні методи.

Перші – це внутрішні методи оцінюють результат кластеризації на основі самих даних без використання будь-якої зовнішньої інформації. Внутрішні методи можуть виміряти, наскільки згуртованими та розділеними є кластери, на основі відстаней або подібності між точками даних. Наприклад, можна використовувати внутрішні методи, щоб оцінити, наскільки компактними та чіткими є кластери та наскільки добре вони відображають природну структуру

даних. Найпоширеніші внутрішні методи включають:

– силуетний коефіцієнт (Silhouette Score). Цей метод вимірює, наскільки об'єкти в кожному кластері подібні один до одного, порівняно з об'єктами інших кластерів. Значення силуетного коефіцієнта лежить у діапазоні від -1 до 1. Ближче до 1 вказує на кращу якість кластеризації;

– індекс Девіса-Болдуїна (Davies-Bouldin Index). Метод оцінює відмінність між кластерами. Низьке значення індексу вказує на кращу кластеризацію, коли кластери відділені один від одного і мають однорідну внутрішню структуру;

– коефіцієнт Калініна-Гловацького (Calinski-Harabasz Index). Метод вимірює якість кластеризації, використовуючи внутрішню та зовнішню дисперсію кластерів. Вище значення коефіцієнта вказує на кращу кластеризацію.

Ці метрики допомагають оцінити якість кластеризації, використовуючи тільки внутрішню структуру даних, без урахування зовнішньої інформації про правильність розбиття на кластери. Вони можуть бути корисні для вибору оптимального числа кластерів та порівняння різних алгоритмів кластеризації.

Зовнішні методи порівнюють вихідні дані кластеризації з деякою зовнішньою або попередньою інформацією, такою як мітки класу, знання домену або основна правда. Зовнішні методи можуть вимірювати, наскільки добре кластери відповідають попередньо визначеним категоріям або критеріям. Наприклад, можна використовувати зовнішні методи, щоб оцінити, наскільки точно алгоритм кластеризації може ідентифікувати різні типи клієнтів, продуктів або документів. Деякі поширені зовнішні методи: чистота, ентропія, індекс Ранда та F-міра.

Відносні методи порівнюють результат кластеризації різних алгоритмів, параметрів або номерів кластерів і вибирають найкращий на основі певних критеріїв. Відносні методи можуть допомогти знайти оптимальне рішення для даних і цілей, тестуючи різні варіанти та ранжуючи їх. Наприклад, можна використовувати відносні методи, щоб визначити, скільки кластерів підходить для наданих даних, або який алгоритм більше підходить для проблеми. Деякі з

поширених відносних методів – метод ліктя, статистика розриву та аналіз стабільності.

Методи стабільності оцінюють стійкість і узгодженість виходу кластеризації за різних умов, таких як збурення даних, варіації вибірки або модифікації алгоритму. Методи стабільності можуть виміряти, наскільки чутливими та надійними є результати кластеризації та наскільки добре вони узагальнюють нові дані. Наприклад, можна використовувати методи стабільності, щоб оцінити, наскільки кластери змінюються, коли додається шум, видаляються викиди або використовуються різні міри відстані. Деякі загальні методи стабільності – кластерний коефіцієнт Жаккара, середня частка неперекриття та середня відстань.

Візуальні методи використовують графічні інструменти або методи для відображення та перевірки результатів кластеризації та отримання розуміння даних і кластерів. Візуальні методи можуть допомогти дослідити розподіл даних, форми кластерів, якість кластерів і зв’язки кластерів. Наприклад, можна використовувати візуальні методи, щоб побачити, як кластери розташовані на точковій діаграмі, дендрограмі або тепловій карті. Деякі поширені візуальні методи – це зменшення розмірності, дерево перевірки кластерів і пошук проекції.

Гібридні методи поєднують два або більше вищевказаних методів, щоб отримати більш повну та збалансовану оцінку результату кластеризації. Гібридні методи можуть використовувати сильні сторони та подолати обмеження різних методів, а також надати більш цілісне та детальне уявлення про якість і валідність кластеризації. Наприклад, можна використовувати гібридні методи для інтеграції зовнішніх і внутрішніх критеріїв, для порівняння та перевірки різних рішень кластеризації або для вдосконалення та вдосконалення візуальних методів. Деякі поширені гібридні методи – це матриця порівняння кластерів, консенсусна кластеризація та інтерактивна кластеризація.

Деякі методи оцінки алгоритму кластеризації будуть використовуватися при аналізу роботи методу кластеризації набору документів для ведення електронного документообігу.

Для аналізу продуктивності алгоритмів можна використати набір з штучно згенерованих даних для отримання необмеженої кількість зразків і можливості систематично змінювати будь-яку з вищезгаданих властивостей набору даних. Це дозволяє широко вимірювати продуктивність алгоритмів кластеризації, а також надає можливість кількісно визначити чутливість продуктивності до невеликих змін у даних.

Кожен алгоритм кластеризації базується на наборі параметрів, які необхідно налаштувати для досягнення бажаної продуктивності, що є важливим моментом, який слід враховувати під час порівняння алгоритмів кластеризації. Проблемою машинного навчання є визначення відповідної процедури для встановлення значень параметрів. Як правило, процедура оптимізації використовується для пошуку конфігурації параметрів, яка забезпечить найкращу продуктивність. Але такий підхід має деякі недоліки.

По-перше, налаштування параметрів на певному наборі даних може призвести до надмірного навчання, тобто конкретні значення, які дають хорошу продуктивність, можуть призвести до низької продуктивності при перегляді нових даних. По-друге, оптимізація параметрів може бути неможлива в деяких випадках через складність багатьох алгоритмів у поєднанні з їх великою кількістю параметрів. Нарешті, дослідники використовують класифікатори або алгоритми кластеризації, використовуючи параметри за замовчуванням, надані програмним забезпеченням. Тому необхідні експерименти, щоб оцінити та порівняти ефективність алгоритмів кластеризації в оптимізованому сценарії та сценарії за замовчуванням.

Таблиця 2.1 - Порівняння методів кластеризації

№ з/п	Характеристика	K-середніх	Ієрархічна кластеризація	DBSCAN
1	Потреба у визначенні кластерів	Так, перед виконанням алгоритму	Ні, кластери формуються під час обробки	Ні, визначає кластери автоматично
2	Ефективність для різних форм кластерів	Кулясті або групи за	Залежить від методу, але	Добре визначає непрості

		геометричними формами	може працювати з будь-якою формою	форми кластерів
3	Сприятливість для великих обсягів даних	Так, ефективний для великих обсягів	Може бути менш ефективним	Може мати складність у виконанні на великих обсягах
4	Вимога до обчислювальних ресурсів	Низька	Залежить від кількості спостережень та глибини дерева	Залежить від параметрів, може бути високою
5	Стійкість до викидів	Чутливий до викидів	Залежить від методу, але може бути менш чутливий	Добре виявляє викиди та шум
6	Складність реалізації	Простий	Може бути складним у великих деревах	Може бути складним для великих наборів даних

Висновки до розділу 2

1. Проведено аналіз існуючих методів кластеризації, що дало змогу систематизувати основні їх переваги і недоліки.

2. Описано запропонований метод кластеризації що використовує модифікований алгоритм k-середніх, що дало змогу виділити запропонований метод серед інших.

3. Розглянуто методики оцінювання ефективності алгоритмів кластеризації, що дало змогу дослідити запропонований метод.

РОЗДІЛ 3 РЕАЛІЗАЦІЯ МЕТОДУ КЛАСТЕРИЗАЦІЇ НАБОРУ ДОКУМЕНТІВ

3.1 Опис засобів реалізації

Для реалізації методу було використано інструменти мови програмування Python, а саме середовище Jupyter, а також бібліотеки Matplotlib та Scikit-learn.

Python – це інтерпретована об’єктно-орієнтована мова програмування високого рівня з динамічною семантикою. Python має репутацію мови, зручної для початківців, замінивши Java як найпоширенішу початкову мову, оскільки вона справляється з більшою частиною складності для користувача, дозволяючи початківцям зосередитися на повному розумінні концепцій програмування, а не на найменших деталях.

Python використовується для веб-розробки на стороні сервера, розробки програмного забезпечення, математики та системних сценаріїв, і популярний для швидкої розробки додатків, а також як мова сценаріїв або з’єднувальна мова для зв’язування існуючих компонентів завдяки своїм високорівневим вбудованим структурам даних, динамічний тип і динамічне зв'язування. Витрати на технічне обслуговування програми знижуються завдяки Python завдяки легкому освоєнню синтаксису та акценту на зручності читання. Крім того, підтримка Python модулів і пакетів полегшує модульні програми та повторне використання коду. Python є мовою спільноти з відкритим вихідним кодом.

Особливості та переваги Python:

- сумісний із різними платформами, включаючи Windows, Mac, Linux, Raspberry Pi та інші;
- використовує простий синтаксис, який можна порівняти з англійською мовою, що дозволяє розробникам використовувати менше рядків, ніж інші мови програмування;
- працює на системі інтерпретатора, яка дозволяє негайно виконувати код, швидко відстежуючи прототипування;

– можна обробляти процедурним, об’єктно-орієнтованим або функціональним способом.

Python, мова з динамічною типізацією, є особливо гнучкою, усуває жорсткі правила створення функцій і пропонує більшу гнучкість вирішення проблем за допомогою різноманітних методів. Python також дозволяє запускати програми, здійснюючи перевірку типів даних під час виконання, а не під час компіляції.

Середовище Jupyter – це серверно-клієнтська програма, яка дозволяє редагувати та запускати документи блокнота через веб-браузер. Програму Jupyter можна запустити на локальному робочому столі, не потребуючи доступу до інтернету, або встановити на віддаленому сервері та отримати доступ через інтернет.

Matplotlib – це кросплатформна бібліотека для візуалізації даних і графічного побудови (гістограми, точкові діаграми, стовпчасті діаграми тощо) для Python і його числового розширення NumPy. Таким чином, він пропонує життєздатну альтернативу MATLAB з відкритим кодом. Розробники також можуть використовувати API matplotlib (інтерфейси прикладного програмування) для вбудовування графіків у програми GUI.

Сценарій matplotlib Python структурований так, що в більшості випадків для створення графіка візуальних даних потрібно лише кілька рядків коду. Рівень сценаріїв matplotlib накладає два API:

– API pyplot — це ієрархія об’єктів коду Python, очолювана matplotlib.pyplot

– набір об’єктів об’єктно-орієнтованого API, який можна зібрати з більшою гнучкістю, ніж pyplot. Цей API забезпечує прямий доступ до базових рівнів Matplotlib.

Scikit-learn – це бібліотека на Python, яка надає багато алгоритмів неконтрольованого та контрольованого навчання. Він створений на основі деяких технологій, з якими ви, можливо, вже знайомі, як-от NumPy, pandas і Matplotlib.

Функціональність, яку надає scikit-learn, включає:

– регресія , включаючи лінійну та логістичну регресію;

– класифікація , включаючи K-найближчих сусідів;
– кластеризація , включаючи K-Means і K-Means++;
– вибір моделі;
– попередня обробка , включаючи нормалізацію мінімум-макс

3.2 Реалізація методу кластеризації набору документів

Для перевірки методу запропоновано використати набір файлів з директорії з описами (синопсисами) фільмів. Використано файли з назвами 100 найкращих фільмів. Запропоновано розділити фільми на 5 кластерів. На початку необхідно завантажити відповідні бібліотеки для роботи:

```
import numpy as np
import pandas as pd
import nltk
import re
import os
import codecs
from sklearn import feature_extraction
import mpld3
```

Далі необхідно вибрати директорію з документами, котрі містять опис фільмів. Нехай назвою файлу будуть назви фільмів, а вмістом файлів буде їх короткий опис ключових елементів сюжету (синопсис), який передає основні події, ідеї та характери фільму. Проводиться пошук усіх файлів та зчитування з них інформації. Формується два списки *'titles'* - назви фільмів та у відповідному до заголовків порядку список *'synopses'* - синопсис фільмів:

```
def analyze_directory(directory_path):
    # Перевіряємо, чи директорія існує
    if not os.path.exists(directory_path):
        print(f"Директорії {directory_path} не існує.")
        return None, None

    # Ініціалізуємо списки для збереження назв файлів та їх вмісту
    titles = []
    synopses = []
```

```
    # Проходимося по всіх файлах у директорії
    for filename in os.listdir(directory_path):
        filepath = os.path.join(directory_path, filename)

        # Перевіряємо, чи є це файл
        if os.path.isfile(filepath):
            # Додаємо назву файлу до списку titles
            titles.append(filename)

            # Зчитуємо вміст файлу та додаємо його до списку synopses
            with open(filepath, 'r', encoding='utf-8') as file:
                file_content = file.read()
                synopses.append(file_content)

    return titles, synopses

# Задаємо шлях до директорії, яку хочемо проаналізувати
directory_path = r'f:\films'

# Викликаємо функцію для аналізу директорії
titles, synopses = analyze_directory(directory_path)
```

Першочергове значення має список 'synopses', а список 'titles' в основному буде використовуватись для маркування. Для роботи зі списками будуть використовуватись їх відповідні індекси. Наприклад, можна вивести назву фільму із індексом 10:

```
print(titles[:10])  # перші 10 назв
```

```
['The Godfather', 'The Shawshank Redemption', "Schindler's List", 'Raging Bull', 'Casablanca',
"One Flew Over the Cuckoo's Nest", 'Gone with the Wind', 'Citizen Kane', 'The Wizard of Oz', 'T
itanic']
```

Або, можна вивести перші 200 символів опис першого фільму:

```
print(synopses[0][:200])  # перші 200 символів у першому описі (для 'The Godfather')
print()
print()
```

```
Plot [edit] [ [ edit edit ] ]
 On the day of his only daughter's wedding, Vito Corleone hears requests in his role as the God
father, the Don of a New York crime family. Vito's youngest son, Michael,
```

Для роботи зі стоп-словами, коренями та токенізацією використано бібліотеку NLTK з якої використовується список англомовних англійських стоп-слів:

```
import nltk
nltk.download('stopwords')
```

```
# завантажуємо англійські зупинні слова з nltk як змінну з назвою 'stopwords'
stopwords = nltk.corpus.stopwords.words('english')

print(stopwords[:10])
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', 'your']
```

Далі імпортується бібліотека Snowball Stemmer що є частиною NLTK (). Здійснюється процес розбивання слів на стебла/основи (Stemming) шляхом обрізання префіксів і суфіксів зі слова, щоб отримати його базову форму, яку називають "основою". Наприклад, при використанні стеммінгу слово "running" може бути зменшено до основи "run", "fishing" до "fish", і так далі. Це допомагає враховувати тільки основний сенс слова і ігнорувати його граматичні відмінності:

```
from nltk.stem.snowball import SnowballStemmer
stemmer = SnowballStemmer("english")
```

Нижче описані дві функції: tokenize_and_stem, що розділяє синопсис на список відповідних слів (токенів), а також позначає кожну лексему та tokenize_only, що позначає лише синопсис. Дані функції використовуються для створення словника, який є важливим у випадку коли потрібно використовувати остебла/основи для алгоритму, щоб пізніше перетворювати їх назад у повні слова для виводу:

```
def tokenize_and_stem(text):
    # розділяємо спочатку на речення, а потім на слова, щоб впевнитися, що пунктуація буде
    # визнана як окремий токен
    tokens = [word for sent in nltk.sent_tokenize(text) for word in nltk.word_tokenize(sent)]
    filtered_tokens = []
    # відфільтровуємо токени, які не містять букв (наприклад, числові токени, сировинну пунктуацію)
    for token in tokens:
        if re.search('[a-zA-Z]', token):
            filtered_tokens.append(token)
    stems = [stemmer.stem(t) for t in filtered_tokens]
    return stems

def tokenize_only(text):
    # розділяємо спочатку на речення, а потім на слова, щоб впевнитися, що пунктуація буде визнана
    # як окремий токен
    tokens = [word.lower() for sent in nltk.sent_tokenize(text) for word in nltk.word_tokenize(sent)]
    filtered_tokens = []
    # відфільтровуємо токени, які не містять букв (наприклад, числові токени, сировинну пунктуацію)
    for token in tokens:
        if re.search('[a-zA-Z]', token):
            filtered_tokens.append(token)
    return filtered_tokens
```

Далі використовуються функції визначення стемінгу/токенізації та токенізації, щоб переглянути список коротких словників, та щоб створити два словники: один з основними словами та один із лише маркерами:

```
#  щоб отримати великий  список словників мржна використати extend
totalvocab_stemmed = []
totalvocab_tokenized = []
for i in synopses:
    allwords_stemmed = tokenize_and_stem(i)  # для кожного елемента в 'synopses', токенізуємо/стемимо
    totalvocab_stemmed.extend(allwords_stemmed)  # розширюємо список 'totalvocab_stemmed'

    allwords_tokenized = tokenize_only(i)
    totalvocab_tokenized.extend(allwords_tokenized)
```

Використовуючи ці два списки створюється pandas DataFrame із словником основ (stem) як індекс рядка індексом і токенізованими словами по стовпцю. Перевага цього полягає в тому, що в такий спосіб забезпечується механізм пошуку в основі з можливість тримати весь токен. Недоліком тут є те, що основи відносяться до токенів як однин до багатьох, наприклад основа 'run' може бути пов’язано з 'ran', 'runs', 'running' тощо:

```
vocab_frame = pd.DataFrame({'words': totalvocab_tokenized}, index = totalvocab_stemmed)
print 'there are ' + str(vocab_frame.shape[0]) + ' items in vocab_frame'

 there are 312209 items in vocab_frame
```

Наступним етапом обчислення показника TF-IDF (term frequency–inverse document frequency), що використовується в задачах аналізу текстів та інформаційного пошуку, де вага/значимість слова пропорційна кількості вживань цього слова у документі, і обернено пропорційна частоті вживання слова у інших документах колекції. Даний показник використовується як один з критеріїв релевантності документа до пошукового запиту, а також при розрахунку міри спорідненості документів при кластеризації.

Щоб отримати матрицю TF-IDF спершу необхідно обчислити кількість входження кожної з основ/terms у кожному із документів, що створить матрицю термінів документа (dtm - document-term matrix).

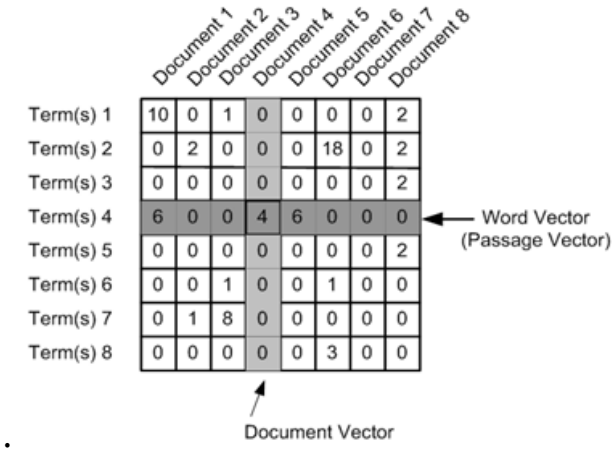

Рисунок 3.1 - Матриця термінів документа.

Далі застосовується частотно-інверсне частотне зважування документа (TF-IDF), де слова, які часто зустрічаються в документі, але не часто в корпусі, отримують вищу вагу, оскільки передбачається, що ці слова містять більше значення по відношенню до документа:

```
from sklearn.feature_extraction.text import TfidfVectorizer

#define vectorizer parameters
tfidf_vectorizer = TfidfVectorizer(max_df=0.8, max_features=200000,
                                   min_df=0.2, stop_words='english',
                                   use_idf=True, tokenizer=tokenize_and_stem, ngram_range=(1,3))

%time tfidf_matrix = tfidf_vectorizer.fit_transform(synopses) #fit the vectorizer to synopses

print(tfidf_matrix.shape)

CPU times: user 29.1 s, sys: 468 ms, total: 29.6 s
Wall time: 37.8 s
(100, 563)
```

Слід зазначити, що параметр max_df: це максимальна частота в документах, яку певна функція може використовуватися в матриці TFI-IDF. Якщо цей термін міститься в більш ніж 80% документів, він мало має значення (у контексті синопсисів фільмів). Параметр min_idf: може бути ціле число і термін повинен бути принаймні в 5 документах, які слід розглядати. Параметр ngram_range: означає, що будуть застосовуватись уніграми, біграми та триграми.

Список terms представляє перелік функцій, які використовуються в матриці TF-IDF. Це словниковий запас:

```
terms = tfidf_vectorizer.get_feature_names()
```

Dist визначається як 1 - косинус подібності кожного документа. Косинусна подібність вимірюється за допомогою матриці TF-IDF і може бути використана для генерування міри подібності між кожним документом та іншими документами в корпусі (кожен синопсис серед синопсисів). Віднімання його від 1 дає косинусну відстань, яку я буду використовувати для побудови на евклідовій (двовимірній) площині. За допомогою dist можна оцінити подібність будь-яких двох або більше синопсисів:

```
from sklearn.metrics.pairwise import cosine_similarity
dist = 1 - cosine_similarity(tfidf_matrix)
print
print
```

Використовуючи матрицю TF-IDF, можна запустити низку алгоритмів кластеризації. K-means ініціалізується заздалегідь визначеною кількістю кластерів (наприклад 5). Кожне спостереження призначається кластеру (кластерне призначення), щоб мінімізувати суму квадратів у межах кластера. Далі обчислюється середнє значення кластерних спостережень і використовується як новий центроїд кластера. Потім спостереження перепризначаються кластерам і центроїдам, які перераховуються в ітераційному процесі, доки алгоритм не досягне збіжності:

```
from sklearn.cluster import KMeans

num_clusters = 5

km = KMeans(n_clusters=num_clusters)

%time km.fit(tfidf_matrix)

clusters = km.labels_.tolist()

CPU times: user 232 ms, sys: 6.64 ms, total: 239 ms
Wall time: 305 ms
```

Для збереження любого об’єкту Python в одному файлі використовуєтсья joblib.dump, щоб зберегти модель, коли вона об’єдналася, і перезавантажити joblib.load модель/перепризначити мітки як кластери:

```
from sklearn.externals import joblib

# щоб зберегти свою модель, розкоментуйте код нижче
# joblib.dump(km,  'doc_cluster.pkl')

km = joblib.load('doc_cluster.pkl')
clusters = km.labels_.tolist()
```

Далі створюється словник назв, рангів, синопсису, призначення кластерів і жанру [ранг і жанр взято з IMDB]. Даний словник конвертується у Pandas DataFrame для легкого доступу:

```
films = {'title': titles, 'rank': ranks, 'synopsis': synopses, 'cluster': clusters, 'genre': genres}

frame = pd.DataFrame(films, index=[clusters], columns=['rank', 'title', 'cluster', 'genre'])

frame['cluster'].value_counts()  # кількість фільмів у кожному кластері (кластери від 0 до 4)
```

```
4    26
0    25
2    21
1    16
3    12
dtype: int64
```

```
grouped = frame['rank'].groupby(frame['cluster'])  # групуємо за кластерами для агрегації

grouped.mean()  # середній ранг (від 1 до 100) для кожного кластера
```

```
cluster
0          47.200000
1          58.875000
2          49.380952
3          54.500000
4          43.730769
dtype: float64
```

Кластери 4 і 0 мають найнижчий ранг, що вказує на те, що вони в середньому містять фільми, які були оцінені як "кращі" у топ-100 списку. Щоб визначити, які з перших n (нехай n=6) слів є найближчими до центроїда кластера. Це дає гарне уявлення про основну тему кластера:

```
from __future__ import print_function

print("Топ-терміни для кожного кластера:")
print()
# сортуємо центри кластерів за близькістю до центроїди
order_centroids = km.cluster_centers_.argsort()[:, ::-1]
```

```
for i in range(num_clusters):
    print("Слова кластера %d:" % i, end='')

    for ind in order_centroids[i, :6]:  # замініть 6 на n слів на кластер
        print(' %s' % vocab_frame.ix[terms[ind].split(' ')].
              values.tolist()[0][0].encode('utf-8', 'ignore'), end=',')
    print()  # додаємо пробіл
    print()  # додаємо пробіл

    print("Назви кластера %d:" % i, end='')
    for title in frame.ix[i]['title'].values.tolist():
        print(' %s,' % title, end='')
    print()  # додаємо пробіл
    print()  # додаємо пробіл

print()
print()
```

```
Top terms per cluster:

Cluster 0 words: family, home, mother, war, house, dies,

Cluster 0 titles: Schindler's List, One Flew Over the Cuckoo's Nest, Gone with the Wind, The Wizard of Oz,
Titanic, Forrest Gump, E.T. the Extra-Terrestrial, The Silence of the Lambs, Gandhi, A Streetcar Named Des
ire, The Best Years of Our Lives, My Fair Lady, Ben-Hur, Doctor Zhivago, The Pianist, The Exorcist, Out of
Africa, Good Will Hunting, Terms of Endearment, Giant, The Grapes of Wrath, Close Encounters of the Third
Kind, The Graduate, Stagecoach, Wuthering Heights,


Cluster 1 words: police, car, killed, murders, driving, house,

Cluster 1 titles: Casablanca, Psycho, Sunset Blvd., Vertigo, Chinatown, Amadeus, High Noon, The French Con
nection, Fargo, Pulp Fiction, The Maltese Falcon, A Clockwork Orange, Double Indemnity, Rebel Without a Ca
use, The Third Man, North by Northwest,


Cluster 2 words: father, new, york, new, brothers, apartments,

Cluster 2 titles: The Godfather, Raging Bull, Citizen Kane, The Godfather: Part II, On the Waterfront, 12
Angry Men, Rocky, To Kill a Mockingbird, Braveheart, The Good, the Bad and the Ugly, The Apartment, Goodfe
llas, City Lights, It Happened One Night, Midnight Cowboy, Mr. Smith Goes to Washington, Rain Man, Annie H
all, Network, Taxi Driver, Rear Window,


Cluster 3 words: george, dance, singing, john, love, perform,

Cluster 3 titles: West Side Story, Singin' in the Rain, It's a Wonderful Life, Some Like It Hot, The Phila
delphia Story, An American in Paris, The King's Speech, A Place in the Sun, Tootsie, Nashville, American G
raffiti, Yankee Doodle Dandy,
```

```
Cluster 4 words: killed, soldiers, captain, men, army, command,

Cluster 4 titles: The Shawshank Redemption, Lawrence of Arabia, The Sound of Music, Star Wars, 2001: A Spa
ce Odyssey, The Bridge on the River Kwai, Dr. Strangelove or: How I Learned to Stop Worrying and Love the
Bomb, Apocalypse Now, The Lord of the Rings: The Return of the King, Gladiator, From Here to Eternity, Sav
ing Private Ryan, Unforgiven, Raiders of the Lost Ark, Patton, Jaws, Butch Cassidy and the Sundance Kid, T
he Treasure of the Sierra Madre, Platoon, Dances with Wolves, The Deer Hunter, All Quiet on the Western Fr
ont, Shane, The Green Mile, The African Queen, Mutiny on the Bounty,


Cluster 4 words: killed, soldiers, captain, men, army, command,

Cluster 4 titles: The Shawshank Redemption, Lawrence of Arabia, The Sound of Music, Star Wars, 2001: A Spa
ce Odyssey, The Bridge on the River Kwai, Dr. Strangelove or: How I Learned to Stop Worrying and Love the
Bomb, Apocalypse Now, The Lord of the Rings: The Return of the King, Gladiator, From Here to Eternity, Sav
ing Private Ryan, Unforgiven, Raiders of the Lost Ark, Patton, Jaws, Butch Cassidy and the Sundance Kid, T
he Treasure of the Sierra Madre, Platoon, Dances with Wolves, The Deer Hunter, All Quiet on the Western Fr
ont, Shane, The Green Mile, The African Queen, Mutiny on the Bounty,
```

Для представлення результатів можна використати багатовимірне масштабування, перетворивши матрицю dist у двовимірний масив:

```python
import os  # для os.path.basename

import matplotlib.pyplot as plt
import matplotlib as mpl

from sklearn.manifold import MDS

MDS()

# конвертуємо дві компоненти, оскільки точки відображаються на двовимірному просторі
# "precomputed" - оскільки надається матриця відстаней
# random_stateє - щоб графік був відтворюваним.
mds = MDS(n_components=2, dissimilarity="precomputed", random_state=1)

pos = mds.fit_transform(dist)  # форма (n_components, n_samples)

xs, ys = pos[:, 0], pos[:, 1]
print()
print()
```

Візуалізацію кластерів документів можна здійснити за допомогою бібліотек matplotlib і mpld3 (оболонка matplotlib для D3.js). Спочатку визначаються словники для переходу від номера кластера до кольору та назви кластера. Назви кластерів визначені на основі слів, які були найближче до кожного центроїда кластера:

```python
# налаштовуємо кольори для кластерів за допомогою словника
cluster_colors = {0: '#1b9e77', 1: '#d95f02', 2: '#7570b3', 3: '#e7298a', 4: '#66a61e'}
```

```
# налаштовуємо назви кластерів за допомогою словника
cluster_names = {0: 'Family, home, war',
                 1: 'Police, killed, murders',
                 2: 'Father, New York, brothers',
                 3: 'Dance, singing, love',
                 4: 'Killed, soldiers, captain'}
```

Далі будуються відпвідні дані (фільми, назви фільмів), розфарбовані за кластером, використовуючи matplotlib:

```
# налаштування ipython для відображення графіків matplotlib
%matplotlib inline

# створюємо фрейм даних, який має результат MDS, а також номери кластерів і назви фільмів
df = pd.DataFrame(dict(x=xs, y=ys, label=clusters, title=titles))

# групуємо за кластерами
groups = df.groupby('label')

# налаштовуємо графік
fig, ax = plt.subplots(figsize=(17, 9)) # задаємо розмір
ax.margins(0.05) # додає 5% заповнення до автомасштабування

# ітеруємо через групи для накладання графіка
# використовуються словники cluster_name та cluster_color викликом
# 'name' для отримання відповідного кольору/позначення

for name, group in groups:
    ax.plot(group.x, group.y, marker='o', linestyle='', ms=12,
            label=cluster_names[name], color=cluster_colors[name],
            mec='none')
    ax.set_aspect('auto')
    ax.tick_params(\
        axis= 'x',          # зміни застосовуються до осі x
        which='both',       # використовуються обидві основні і додаткові мітки
        bottom='off',       # мітки вздовж нижнього краю вимкнені
        top='off',          # мітки вздовж верхнього краю вимкнені
        labelbottom='off')

    ax.tick_params(\
        axis= 'y',          # зміни застосовуються до осі y
        which='both',       # використовуються обидві основні і додаткові мітки
        left='off',         # мітки вздовж лівого краю вимкнені
        top='off',          # мітки вздовж верхнього краю вимкнені
        labelleft='off')

ax.legend(numpoints=1)  # відображення легенди з лише 1 точкою
```

```
# додаємо мітку в позиції x, y із позначенням у якості назви фільму
for i in range(len(df)):
    ax.text(df.ix[i]['x'], df.ix[i]['y'], df.ix[i]['title'], size=8)


plt.show() # відображення графіку

# при необхідності можна зберегти графік
# plt.savefig('clusters_small_noaxes.png', dpi=200)
```

На рисунку 3.2 представлено 5 кластерів розфарбованих у відповідні кольори. Розміщення кожного із фільмів позначено окремою зафарбованою у відповідний кластер точкою. Даний результат дозволяє дослідити розташування одного фільму відносно інших у двовимірному просторі. Близькість в даному випадку еквівалентна схожості, визначеній багатовимірним масштабуванням косинусної відстані між синопсисами, що містяться у матриці TF-IDF. Запропонований метод дозволив на основі зібраних синопсисів відобразити кожен фільм щодо його схожості до всіх інших фільмів. Схожість фільмів в найбільшому залежить від правильно описаних синопсів.

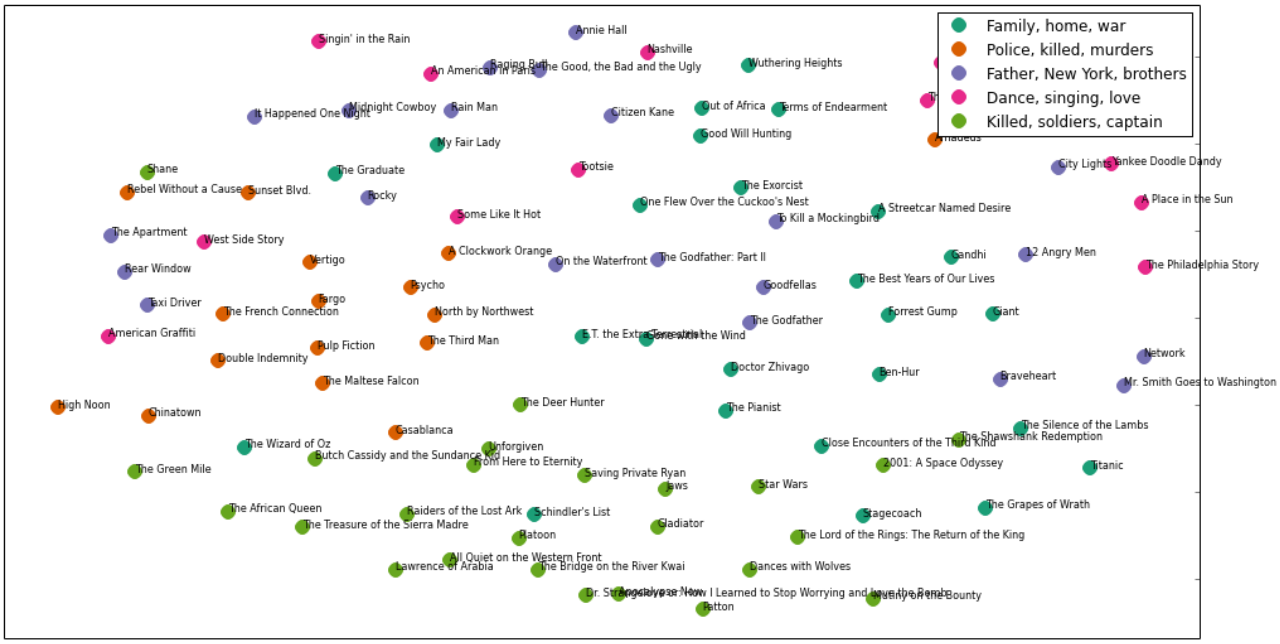

Рисунок 3.2 – Розподіл наявних фільмів по п’ятьом кластерам

3.3 Оцінювання ефективності методу кластеризації набору документів

Оцінка ефективності результатів кластеризації, відома як оцінка кластеризації, має важливе значення для успіху програм кластеризації. Це гарантує, що алгоритм кластеризації визначив значущі кластери в даних, а не лише артефакти випадкового шуму. Крім того, його можна використовувати для визначення, який алгоритм кластеризації найкраще підходить для певного набору даних і завдання, а також для налаштування гіперпараметрів цих алгоритмів

Оцінка якості методу кластеризації набору документів для ведення електронного документообігу вимагає вивчення та визначення ключових параметрів, важливих для функціональності самого методу.

Якщо розглянути питання детальніше, то можна виявити такі показники:

- кількість вершин;
- збіжність вузлів;
- швидкість алгоритму.

Ці функції максимізують покращення в рамках модифікованого алгоритму кластеризації пошукових запитів користувачів порівняно з базовим алгоритмом.

Для дотримання експериментальної методології було відібрано вибірку з 100 документів, а необхідну кількість кластерів було встановлено вручну. Зміна результатів між прогонами є першим кроком в алгоритмі k-середніх. Ці експерименти використовували вимірювання відстані та стандартні представлення, описані у другому розділі. Для аналізу результату було застосовано алгоритм k-середніх та удосконалений алгоритм k-середніх. Алгоритм кластеризації k-середніх дає інший результат, оскільки центроїди обираються випадковим чином, тоді як у покращеному алгоритмі k-середніх значення однакове для кожного виконання, оскільки центроїди прогнозуються вручну.

Значено точності, запам'ятовування та f-міри для існуючого та запропонованого алгоритму (див.рисунок 3.3). F-міра має більше значення для запропонованого алгоритму порівняно з існуючим алгоритмом. Також значення

точності та запам'ятовування є кращими для запропонованого алгоритму порівняно з існуючим алгоритмом (таблиця 3.1).

Таблиця 3.1 - Існуючий та запропонований алгоритм

	Точність	Запам’ятовування	f-міра
k-середнє	0.54474	0.47201	0.42317
ik-середнє	0.8177	0.82	0.818

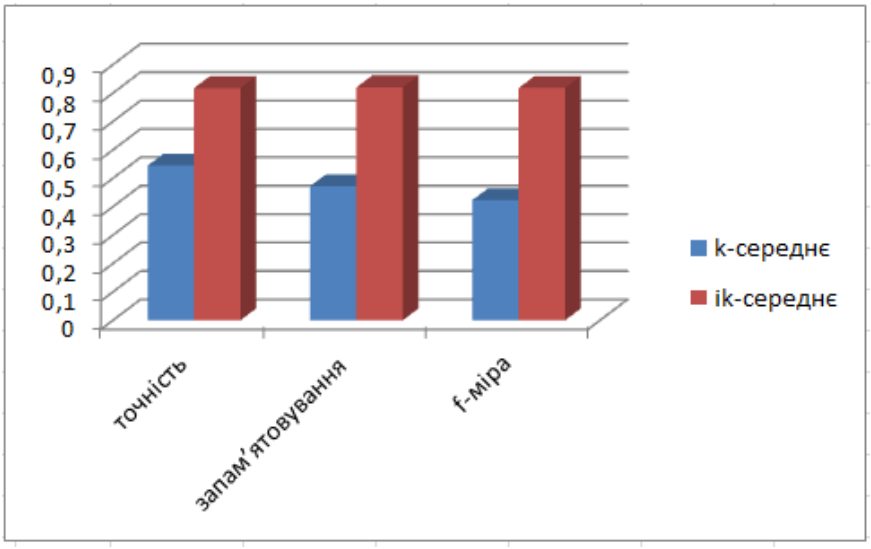

Рисунок 3.3 – Міра точності

На рисуноку 3.4 порівняно два алгоритма за часом. Існуючий алгоритм займає більше часу, ніж запропонований.

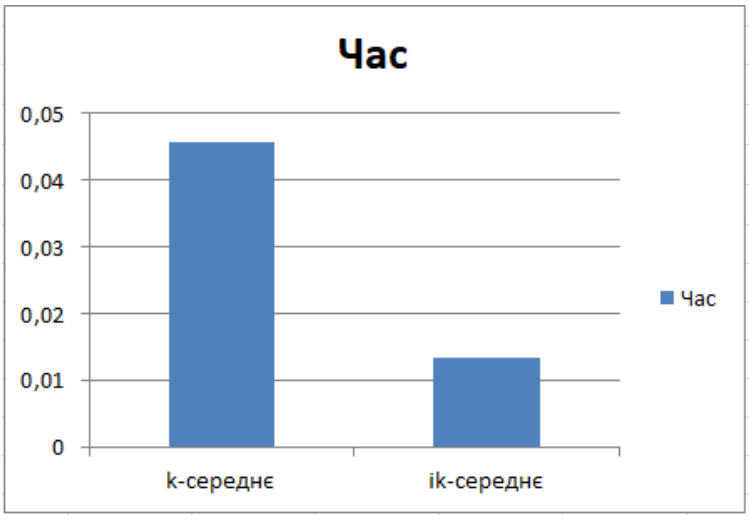

Рисунок 3.4 – Порівняння часу

Кластеризація документів широко використовується для відокремлення схожих документів і несхожих разом. Традиційний алгоритм k-середніх добре працює з певними документами, а центроїди обираються випадково. У запропонованому алгоритмі центроїди прогнозуються вручну. Кожного разу результати збігаються. Експериментальні результати показали, що покращений алгоритм k-середніх працює краще, ніж існуючий алгоритм з точки зору точності, f- міри та часу.

Висновки до розділу 3

1. Проведено опис засобів для реалізації запропонованого методу кластеризації набору документів, що дало змогу виділити їх основні особливості для реалізації поставленої задачі.
2. Представлено реалізацію методу кластеризації набору документів, що містить модифікований метод k-середніх. Це дало змогу розділяти документи по заданих кластерах.
3. Проведено оцінювання ефективності методу кластеризації набору документів, що дало змогу підтвердити ефективність застосування даного методу.

ВИСНОВКИ

1. Проведено аналіз предметної області, основних елементів та аспектів внутрішнього та зовнішнього електронного документообігу. Проаналізовано системи керування документами та інструменти підтримки ефективності даного процесу. Це дало змогу виділити основні елементи та інструменти необхідні для електронного документообігу.

2. Здійснено опис задачі кластеризації, проаналізовано сфери застосування процесу кластеризації, що дало змогу визначити основні критерії для визначення кластерів у відповідних областях даних.

3. Проведено огляд літератури та основних елементів процесу кластеризації, що дало змогу сформувати основні напрямки дослідження та підтвердити актуальність теми.

4. Проаналізовано вимоги до процедури кластеризації, що дало можливість сформувати вимоги до розроблюваного методу кластеризації документів.

5. Проведено аналіз існуючих методів кластеризації, що дало змогу систематизувати основні їх переваги і недоліки.

6. Описано запропонований метод кластеризації що використовує модифікований алгоритм k-середніх, що дало змогу виділити запропонований метод серед інших.

7. Розглянуто методики оцінювання ефективності алгоритмів кластеризації, що дало змогу дослідити запропонований метод.

8. Проведено опис засобів для реалізації запропонованого методу кластеризації набору документів, що дало змогу виділити їх основні особливості для реалізації поставленої задачі.

9. Представлено реалізацію методу кластеризації набору документів, що містить модифікований метод k-середніх. Це дало змогу розділяти документи по заданих кластерах.

10. Проведено оцінювання ефективності методу кластеризації набору документів, що дало змогу підтвердити ефективність застосування даного методу.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. SK Card, GG Robertson, W. York, The WebBook and the Web Forager: An Information Workspace for the World-Wide Web, Proceedings of the ACM/SIGCHI Conference on Human Factors in Computing Systems, Vancouver, 2016, pp. 111–119 .

2. The use of hierarchic clustering in information retrieval. URL: https://www.sciencedirect.com/science/article/abs/pii/0020027171900519

3. Е. Расмуссен, Алгоритми кластеризації, в: WB Frakes, R. Baeza-Yates (Eds.), Information Retrieval, Data Structures and Algorithms, Prentice-Hall, Englewood Cliffs, NJ, 2019, pp. 419–442.

4. M.A. Hearst, J.O. Pedersen, Reexamining the cluster hypothesis: scatter/gather on retrieval results, Proceedings of the 19th Annual International ACM Conference on Research and Development in Information Retrieval, Zurich, 2015, pp. 76–84.

5. M. Sahami, S. Yusufali, Q.W. Baldonado, SONIA: A service for organizing networked information autonomously, Proceeding of the 3rd ACM International Conference on Digital Libraries, Pittsburgh, PA, 2020, pp. 237–246.

6. O. Zamir, O. Etzioni, O. Madani, R.M. Karp, Fast and intuitive clustering of Web documents, Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, 2018, pp. 287–290.

7. Закон України “Про електронні документи та електронний документообіг” № 55 від 17.01.2018 [Електронний ресурс] / Офіційний сайт Верховної ради України. веб-сайт. URL: https://zakon.rada.gov.ua. (дата звернення: 15.01.2022).

8. UC Irvine Machine Learning Repository. URL: https://archive.ics.uci.edu/.

9. What methods can you use to evaluate the performance of a clustering algorithm? 2023 [Електронний ресурс]. URL: https://www.linkedin.com/advice/1/what-methods-can-you-use-evaluate-performance-clustering.

10. ДСТУ 3008:2015 — «Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання» [Чинний від 2017-07-01]. Київ, 2016. 31 с. (Інформація та документація).

11. The Ultimate Guide to Electronic Document Management Systems (EDMS) [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: https://www.make.com/en/blog/edms.

12. Electronic Document Circulation [Електронний ресурс] – Режим доступу до ресурсу: https://pentacomp.net/products/electronic-document-circulation.

13. Archive Automate archiving business processes with a versatile system of storing electronic documents which guarantees a safe, controlled, managed store, easy search, and processing of any kind of document format [Електронний ресурс] – Режим доступу до ресурсу: https://softxpansion.global/products/softxspace-archive.

14. Electronic Document Management [Електронний ресурс] – Режим доступу до ресурсу: https://almexoft.com/areas-of-use/electronic-document-management.

15. Electronic document workflow [Електронний ресурс] – Режим доступу до ресурсу: https://primesoft.biz/electronic-document-workflow/.

16. Electronic document management system [Електронний ресурс] – Режим доступу до ресурсу: https://www.aps-smart.com/en/electronic-document-management-system/.

17. André Miguel Fernandes Rodrigues. Document Clustering as an aPproach to Template Extraction / André Miguel Fernandes Rodrigues., 2021. – 43 c.

18. Electronic circulation of documents [Електронний ресурс] – Режим доступу до ресурсу: https://intratel.pl/en/offer/solutions/electronic-circulation-of-documents/.

19. Electronic Document And Record Management System: A Brief Guide [Електронний ресурс] – Режим доступу до ресурсу: https://www.docupile.com/electronic-document-and-record-management-system-a-brief-guide/.

20. What is Scikit-Learn? [Електронний ресурс] – Режим доступу до ресурсу: https://www.codecademy.com/article/scikit-learn.

21. Neri Van Otten. Top 6 Most Popular Text Clustering Algorithms And How They Work Explained [Електронний ресурс] / Neri Van Otten – Режим доступу до ресурсу: https://spotintelligence.com/2023/01/17/text-clustering-algorithms/.

22. Python Built in Functions. [Електронний ресурс]. URL: https://www.w3schools.com/python/python_ref_functions.asp.

23. Python Matplotlib. Matplotlib Tutorial [Електронний ресурс] – Режим доступу до ресурсу: https://www.w3schools.com/python/matplotlib_intro.asp.

24. JavaScript Get Date Methods. [Електронний ресурс]. URL: https://www.w3schools.com/js/js_date_methods.asp.

25. Clustering algorithms: A comparative approach. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6333366/.

26. Customer Segmentation with Clustering Algorithms URL: https://medium.com/mbektas/customersegmentation-with-clustering-algorithms-in-python-be2e021035a.

27. Michael Steinbach. A Comparison of Document Clustering Techniques [Електронний ресурс] / Michael Steinbach, George Karypis, Vipin Kumar – Режим доступу до ресурсу: https://www.stat.cmu.edu/~rnugent/PCMI2016/papers/DocClusterComparison.pdf.

28. Aggarwal CC, Reddy CK. Data Clustering: Algorithms and Applications. vol. 2 1st ed Chapman & Hall/CRC; 2013.

29. What is Document Clustering Analysis? [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://www.tutorialspoint.com/what-is-document-clustering-analysis.

30. Dmitri G Roussinov. Document clustering for electronic meetings: an experimental comparison of two techniques [Електронний ресурс] / Dmitri G Roussinov, Hsinchun Chen – Режим доступу до ресурсу: https://www.sciencedirect.com/science/article/pii/S0167923699000378.

31. Щодо державної політики підтримки розвитку інноваційних кластерів у промисловості України: Аналітична записка [Електронний ресурс]. – Режим доступу: http://www.niss.gov.ua/ article/.

32. Machine Learning [Електронний ресурс] – Режим доступу до ресурсу: https://www.w3schools.com/ai/default.asp.

33. NumPy Tutorial [Електронний ресурс] – Режим доступу до ресурсу: https://www.w3schools.com/python/numpy/default.asp.

34. Document clustering [Електронний ресурс] – Режим доступу до ресурсу: https://www.knime.com/nodeguide/other-analytics-types/text-processing/document-clustering.

35. Improvement of K-means Cluster Quality by Post Processing Resulted Clusters [Електронний ресурс] // Procedia Computer Science. – 2022. – Режим доступу до ресурсу: https://www.sciencedirect.com/science/article/pii/S1877050922000096.

36. An Improved K-Means Algorithm Based on Evidence Distance [Електронний ресурс] / Ailin Zhu, Zexi Hua, Yongchuan Tang, Lingwei Miao // Entropy. – 2021. – Режим доступу до ресурсу: https://www.mdpi.com/1099-4300/23/11/1550.

37. Mins Read. What is Clustering in Machine Learning: Types and Methods [Електронний ресурс] / MINS READ // Analytixlabs. – 2022. – Режим доступу до ресурсу: https://www.analytixlabs.co.in/blog/types-of-clustering-algorithms/.

38. Kanwal Garg. Document Clustering using Improved K-means Algorithm [Електронний ресурс] / Kanwal Garg // Academia. – 2016. – Режим доступу до ресурсу: https://www.academia.edu/83145459/Document_Clustering_using_Improved_K_means_Algorithm.

39. Aryan Garg. Clustering Unleashed: Understanding K-Means Clustering [Електронний ресурс] / Aryan Garg // KDnuggets. – 2023. – Режим доступу до ресурсу: https://www.kdnuggets.com/2023/07/clustering-unleashed-understanding-kmeans-clustering.html.

40. Data Science Tutorial [Електронний ресурс] – Режим доступу до ресурсу: https://www.w3schools.com/datascience/default.asp.

41. Clustering text documents using k-means [Електронний ресурс] – Режим доступу до ресурсу: https://scikit-learn.org/stable/auto_examples/text/plot_document_clustering.html.

42. Штовба С.Д., Козачко О.М. Machine Learning: стартовий курс: електронний навчальний посібник. – Вінниця: Вінницький національний технічний університет, 2020. – 81 с.

43. Способи прискорення обчислення матриці кореляції термів в методі острівної кластеризації текстів / Юсин Я.О., Заболотня Т.М. // Прикладна математика та комп'ютинг ПМК-2018 : Тез. доп. – Київ, 21- 23 березня 2018. – С.272–276.

44. Document Clustering with Python [Електронний ресурс] – Режим доступу до ресурсу: http://brandonrose.org/clustering.

45. Gan G, Ma C, Wu J. Data Clustering: Theory, Algorithms, and Applications[M]. 2nd. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2020.

46. Managing Jupyter Kernels in JupyterLab [Електронний ресурс] Posit Workbench User Guide. – 2022. – Режим доступу до ресурсу: https://docs.posit.co/ide/server-pro/user/2023.03.1/jupyter-lab/guide/jupyter-kernel-management.html.

47. Carbonneau M.A. Multiple instance learning: A survey of problem characteristics and applications / M.-A. Carbonneaua, V. Cheplygina, E. Grangera, G. Gagnon // Pattern Recognition. – 2018. – Vol. 77. – P. 329–353.

48. Чумадевська Х.В., Загородня Д.І. Кластеризація набору документів для ведення електронного документообігу. VIII Науково-практична конференція молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі». 2023. С. 68.

49. Чумадевська Х.В., Загородня Д.І. Метод кластеризації документів вдосконаленим методом k-середніх. Збірник тези доповідей міжнародної науково-

практичної інтернет-конференції «Інформаційне суспільство: технологічні, економічні та технічні аспекти становлення». 2023 – Вип. 83. С. 00-00. http://www.konferenciaonline.org.ua/ua/article/id-1521/.

50. Загальні рекомендації з підготовки, оформлення, захисту та оцінювання випускних кваліфікаційних робіт здобувачів вищої освіти першого «бакалаврського» і другого «магістерського» рівнів / За ред. доц. М.І. Шинкарика. Тернопіль: ТНЕУ, 2018. 67 с.

51. Комар М.П., Саченко А.О., Васильків Н.М. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп’ютерні науки» спеціальності 122 «Комп’ютерні науки» за другим (магістерським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2021. 32 с.

ДОДАТОК А

Сертифікат регіональної цифрової трансформації

ДОДАТОК Б

Копії публікацій

ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

ЗБІРНИК ПРАЦЬ

VIII НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ МОЛОДИХ ВЧЕНИХ І СТУДЕНТІВ «ІНТЕЛЕКТУАЛЬНІ КОМП'ЮТЕРНІ СИСТЕМИ ТА МЕРЕЖІ»

5 ГРУДНЯ 2023

KI.WUNU.EDU.UA/CONFERENCE/

ТЕРНОПІЛЬ

2023

Чумадевська Х.В.
магістрантка 2 курсу ФКІТ ЗУНУ
Науковий керівник к.т.н. Загородня Д.І., кафедра ІОСУ ЗУНУ

КЛАСТЕРИЗАЦІЯ НАБОРУ ДОКУМЕНТІВ ДЛЯ ВЕДЕННЯ ЕЛЕКТРОННОГО ДОКУМЕНТООБІГУ

Вступ. Більшість країн, включаючи і Україну, ведуть політику діджиталізації, що визначається активним впровадженням цифрових технологій. Необхідність переходу до електронного документообігу стає необхідною для більшості організацій і підприємств. З інтенсифікацією генерації документів та інформації виникає потреба в систематизації цього потоку. Тому, кластеризація документів є важливою складовою для забезпечення ефективного управління та зручного доступу до цієї інформації.

Постановка задачі. Об'єкт дослідження – процес класифікації документів в електронному документообігу з використанням методів кластеризації. Предмет дослідження – методи класифікації та кластеризації набору документів, які використовуються для ведення електронного документообігу. Мета дослідження – вдосконалення методів класифікації документів для оптимізації електронного документообігу.

Основний матеріал. Кластеризація набору документів дозволяє групувати схожі документи разом, спрощуючи їх пошук та каталогізацію. Попередні наукові роботи та дослідження в галузі кластеризації документів охоплюють різні методи та підходи. Найпоширеніші метод: ієрархічна кластеризація, метод k-середніх (K-Means), DBSCAN, Spectral Clustering та інші [1-3]. Кожен з цих методів має свої переваги та обмеження. Наприклад, ієрархічна кластеризація дозволяє створювати ієрархію кластерів, що корисно для вивчення структури даних на різних рівнях деталізації, проте вона обчислювально-затратна для великих наборів даних [1]. Метод k-середніх, навпаки, є швидким та ефективним методом, але вимагає передбачення кількості кластерів перед виконанням алгоритму[2].

Запропонований метод базується на використанні ряду алгоритмів та підходів, які дозволяють групувати документи за схожістю та виявляти структурні та тематичні зв'язки між ними. Перший крок – це збір та підготовка даних. Отримати доступ до набору документів, які потрібно кластеризувати, і перевірити їх на наявність необхідних метаданих, таких як заголовки, дати створення, автори тощо. У разі необхідності документи потрібно конвертувати у текстовий формат для подальшої обробки. Наступним кроком є витягування ключових слів та векторна репрезентація документів. Для поліпшення розуміння змісту документів використовуються алгоритми обробки тексту для витягування ключових слів та фраз з кожного документа. Ці ключові слова служать для створення векторної репрезентації документів, яка дозволяє порівнювати їх за схожістю, тобто розбити на кластери. Після кластеризації проводиться оцінка та валідація результатів використовуючи різні метрики, такі як індекс Силуета. Цей етап дозволить визначити оптимальну кількість кластерів та переконатися, що результати відповідають поставленим цілям.

Висновки. Проведення класифікації та кластеризації електронних документів спрощує організацію ведення електронного документообігу, виконуючи групування документів на основі схожості. Це підвищує продуктивність працівників, дозволяючи ефективніше працювати з документами, а також уникати помилок та забезпечити надійність управління.

Список літератури

1.Chaitanya Reddy Patlolla. Understanding the concept of Hierarchical clustering Technique. Published in Towards Data Science. – 2018. – URL: https://towardsdatascience.com/understanding-the-concept-of-hierarchical-clustering-technique-c6e8243758ec.

2.Clustering Algorithms. – URL: https://developers.google.com/machine-learning/clustering/clustering-algorithms.

3.Li Jingnan, Lin Chuan, Huang Ruizhang, Qin Yongbin, Chen Yanping. Intention-guided deep semi-supervised document clustering via metric learning. – Journal of King Saud University - Computer and Information Sciences. – Vol. 35, Iss.1, 2023, P. 416-425.

МЕТОД КЛАСТЕРИЗАЦІЇ ДОКУМЕНТІВ ВДОСКОНАЛЕНИМ МЕТОДОМ K-СЕРЕДНІХ

07.12.2023 22:59

[1. Інформаційні системи і технології]

Автор: **Чумадевська Христина Василівна, студентка, Західноукраїнський національний університет, м. Тернопіль; Загородня Діана Іванівна, кандидат технічних наук, доцент, Західноукраїнський національний університет, м. Тернопіль**

ORCID: 0000-0002-9764-3672 Diana Zahorodnia

Системи управління документами (DMS) – це системи, які пропонують такі послуги, як зберігання, керування версіями, метаданими, безпека, а також можливості індексування та пошуку. Велика кількість документів може бути автоматично згрупована в класи документів, які містять схожу інформацію. Для цього застосовують методи кластеризації.

Елементами DMS для документів є: інтеграція, вилучення метаданих, збір, перевірка, індексування, зберігання, пошук, розповсюдження, безпека, робочий процес, співпраця, керування версіями, пошук, публікація та відтворення.

Процес кластеризації документів складається з декількох етапів. Спочатку виконується попередня обробка наборів даних, які надають набір токенів для моделі векторного простору (VSM). VSM – це процес пошуку, який працює за моделлю Tf-Idf. Для обчислення відстані між різними кластерами використовуються міри подібності.

У роботі для кластеризації документів використовується дерево подібності документів, яке виокремлює послідовність фраз і слів у документах. Для кластеризації сегментів на основі схожості використовується підхід побудови кластерів розташованих на максимально великій відстані [1]. Алгоритм генетичної кластеризації використовується для вирішення проблеми агрегації кластерів. Процес кластеризації зображений на рисунку 1.

Попередня обробка виконується над звичайними текстовими документами і генерує набір токенів для виводу в VSM. Ця методика забезпечує оптимальну якість кластерів. Основні етапи препроцесорної обробки полягають у наступному:

1. Фільтрація: для видалення розділових знаків і спеціальних символів.

2. Токенізація: для розбиття токенів на окремі слова та токени.

3. Зупинити видалення слів: слова, що не мають значення, видаляються.

4. Стеммінг: утворюється основна форма слів.

5. Обрізка: для видалення низькочастотних слів.

Рисунок 1 – Процес кластеризації документів

Пошук термінів знаходить ексклюзивні терміни з кожної доступної категорії. Кожному терміну присвоюється порогове значення як вага. Частота терміна tf (i, j) – це кількість разів, коли термін і зустрічається в документі j. Якщо частота терміна tf більша за порогове значення, то значення додається, інакше відхиляється.

Вилучення функцій використовується для видалення набору ключових слів з документів. VSM – це техніка пошуку в інтелектуальному аналізі даних, також відома як модель «частота терміна - частота документа». Це стандартна алгебраїчна модель представлення тексту. Кожен документ представляється у вигляді n-вимірного вектора за допомогою вектора ознак. Значення кожного елемента вектора відображає важливість відповідної ознаки в документі. За допомогою цієї моделі схожість між документами вимірюється шляхом обчислення відстані між векторами документів. Якщо документи містять однакові ключові слова, вони вважаються схожими. Частота терміна нормалізується відносно максимальної частоти всіх термінів, що зустрічаються в документі.

Також обчислюються евклідова схожість, оцінка ефективності кластеризації (вимірюється за допомогою F-міри), точність (визначається як відношення кількості позитивних результатів до кількості позитивних результатів плюс кількість хибних результатів), відгук (визначається як відношення кількості істинно позитивних спрацьовувань до кількості істинно позитивних спрацьовувань плюс кількість помилково негативних результатів).

Наступний крок – це покращений метод k-середніх. Удосконалена кластеризація за методом k-середніх використовує алгоритм на основі розбиття. Одним з таких алгоритмів є Bisecting K Means Methods [2], який починає з розбиття всієї множини точок на два кластери, вибирає один із них, поділяє його, а потім повторює цей процес, поки не створить k кластерів. Гібридна бісектриса k-середніх використовує комбінацію бісектриси k-середніх та ієрархічного алгоритму розбиття для отримання оптимальних кластерів. Цей удосконалениий алгоритм спрямований на автоматичну кластеризацію та усунення недоліків методу K-Means [3].

Робота виконується на наборі даних mini_newsgroups [4]. Для порівняння класифікація виконувалась методом k-середніх та покращеним методом k-середніх на 300 документах з mini_Newsgroup. Результати представлені на рисунку 2.

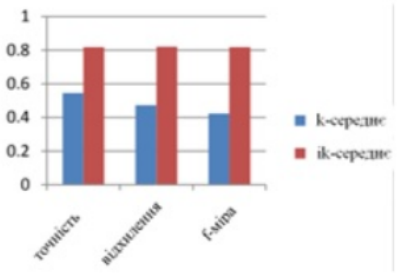

Рисунок 2 – Результати класифікації документів методом k-середніх та удосконаленим методом

F-міра має більше значення для запропонованого алгоритму порівняно з існуючим алгоритмом. Також значення точності та відхилення є кращими для запропонованого алгоритму порівняно з існуючим алгоритмом.

На рисунку 3 представлено порівняння цих методів за часом. Існуючий метод потребує більшого часу виконання, ніж запропонований.

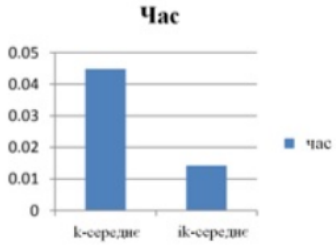

Рисунок 3 – Порівняння методів за часом виконання

В умовах інтенсивної генерації документів, кластеризація стає необхідним інструментом для структурування, управління та зручного доступу до інформації, а також може служити основою для впровадження різноманітних технологій аналізу та автоматизації обробки даних. Традиційний алгоритм k-середніх добре працює з певними документами, а центроїди обираються випадково. У запропонованому алгоритмі центроїди прогнозуються вручну. Експериментальні результати показали, що покращений алгоритм k-середніх працює краще, ніж існуючий алгоритм з точки зору точності, f- міри та часу.

Література

1. Rupesh Kumar Mishra, Knika Sain, Sakshi Bagri, «Text Document Clustering On The Basis Of Inter Passage Approach By Using K-Means», International Conference On Computing, Communication And Automation,(ICCCA-2015), may 15-16, pp:110-113,IEEE, 2018.

2. Pradeep Rai. Shubha Singh, «A Survey Of Clustering Techniques», International Journal Of Computer Applications, volume 7,pp:1-5, 2020.

3. Improvement of K-means Cluster Quality by Post Processing Resulted Clusters. URL: https://www.sciencedirect.com /science/article/pii/S1877050922000096, 2022 .

4. UC Irvine Machine Learning Repository. URL: https://archive.ics.uci.edu/.

ДОДАТОК В

Код реалізації методу

```
import numpy as np
import pandas as pd
import nltk
import re
import os
import codecs
from sklearn import feature_extraction
import mpld3
```

```
def analyze_directory(directory_path):
    # Перевіряємо, чи директорія існує
    if not os.path.exists(directory_path):
        print(f"Директорії {directory_path} не існує.")
        return None, None

    # Ініціалізуємо списки для збереження назв файлів та їх вмісту
    titles = []
    synopses = []

    # Проходимося по всіх файлах у директорії
    for filename in os.listdir(directory_path):
        filepath = os.path.join(directory_path, filename)

        # Перевіряємо, чи є це файл
        if os.path.isfile(filepath):
            # Додаємо назву файлу до списку titles
            titles.append(filename)

            # Зчитуємо вміст файлу та додаємо його до списку synopses
            with open(filepath, 'r', encoding='utf-8') as file:
                file_content = file.read()
                synopses.append(file_content)

    return titles, synopses
```

```
# Задаємо шлях до директорії, яку хочемо проаналізувати
directory_path = r'f:\films'

# Викликаємо функцію для аналізу директорії
titles, synopses = analyze_directory(directory_path)

print(titles[:10])  # перші 10 назв

print(synopses[0][:200])  # перші 200 символів у першому описі (для 'The Godfather')
print()
print()
```

```
import nltk
nltk.download('stopwords')
# завантажуємо англійські зупинні слова з nltk як змінну з назвою 'stopwords'
stopwords = nltk.corpus.stopwords.words('english')

print(stopwords[:10])
```

```
from nltk.stem.snowball import SnowballStemmer
stemmer = SnowballStemmer("english")
```

```
def tokenize_and_stem(text):
    # розділяємо спочатку на речення, а потім на слова, щоб впевнитися, що пунктуація буде
    # визнана як окремий токен
    tokens = [word for sent in nltk.sent_tokenize(text) for word in nltk.word_tokenize(sent)]
    filtered_tokens = []
    # відфільтровуємо токени, які не містять букв (наприклад, числові токени, сировинну пунктуацію)
    for token in tokens:
        if re.search('[a-zA-Z]', token):
            filtered_tokens.append(token)
    stems = [stemmer.stem(t) for t in filtered_tokens]
    return stems

def tokenize_only(text):
    # розділяємо спочатку на речення, а потім на слова, щоб впевнитися, що пунктуація буде визнана
    # як окремий токен
    tokens = [word.lower() for sent in nltk.sent_tokenize(text) for word in nltk.word_tokenize(sent)]
    filtered_tokens = []
    # відфільтровуємо токени, які не містять букв (наприклад, числові токени, сировинну пунктуацію)
    for token in tokens:
        if re.search('[a-zA-Z]', token):
            filtered_tokens.append(token)
    return filtered_tokens
```

```
#  щоб отримати великий  список словників мржна використати extend
totalvocab_stemmed = []
totalvocab_tokenized = []
for i in synopses:
    allwords_stemmed = tokenize_and_stem(i)  # для кожного елемента в 'synopses', токенізуємо/стемимо
    totalvocab_stemmed.extend(allwords_stemmed)  # розширюємо список 'totalvocab_stemmed'

    allwords_tokenized = tokenize_only(i)
    totalvocab_tokenized.extend(allwords_tokenized)
```

```
vocab_frame = pd.DataFrame({'words': totalvocab_tokenized}, index = totalvocab_stemmed)
print 'there are ' + str(vocab_frame.shape[0]) + ' items in vocab_frame'
```

```
from sklearn.feature_extraction.text import TfidfVectorizer

#define vectorizer parameters
tfidf_vectorizer = TfidfVectorizer(max_df=0.8, max_features=200000,
                                   min_df=0.2, stop_words='english',
                                   use_idf=True, tokenizer=tokenize_and_stem, ngram_range=(1,3))

%time tfidf_matrix = tfidf_vectorizer.fit_transform(synopses) #fit the vectorizer to synopses

print(tfidf_matrix.shape)
```

```
terms = tfidf_vectorizer.get_feature_names()
```

```
from sklearn.metrics.pairwise import cosine_similarity
dist = 1 - cosine_similarity(tfidf_matrix)
print
print

from sklearn.cluster import KMeans

num_clusters = 5

km = KMeans(n_clusters=num_clusters)

%time km.fit(tfidf_matrix)

clusters = km.labels_.tolist()
from sklearn.externals import joblib

# щоб зберегти свою модель, розкоментуйте код нижче
# joblib.dump(km,  'doc_cluster.pkl')

km = joblib.load('doc_cluster.pkl')
clusters = km.labels_.tolist()
films = {'title': titles, 'rank': ranks, 'synopsis': synopses, 'cluster': clusters, 'genre': genres}

frame = pd.DataFrame(films, index=[clusters], columns=['rank', 'title', 'cluster', 'genre'])

frame['cluster'].value_counts()  # кількість фільмів у кожному кластері (кластери від 0 до 4)

grouped = frame['rank'].groupby(frame['cluster'])  # групуємо за кластерами для агрегації

grouped.mean()  # середній ранг (від 1 до 100) для кожного кластера
from __future__ import print_function

print("Топ-терміни для кожного кластера:")
print()
# сортуємо центри кластерів за близькістю до центроїди
order_centroids = km.cluster_centers_.argsort()[:, ::-1]

for i in range(num_clusters):
    print("Слова кластера %d:" % i, end='')

    for ind in order_centroids[i, :6]:  # замініть 6 на n слів на кластер
        print(' %s' % vocab_frame.ix[terms[ind].split(' ')].
              values.tolist()[0][0].encode('utf-8', 'ignore'), end=',')
    print()  # додаємо пробіл
    print()  # додаємо пробіл
```

```
    print("Назви кластера %d:" % i, end='')
    for title in frame.ix[i]['title'].values.tolist():
        print(' %s,' % title, end='')
    print()  # додаємо пробіл
    print()  # додаємо пробіл

print()
print()
```

Модуль графічного представлення результатів

```
import os  # для os.path.basename

import matplotlib.pyplot as plt
import matplotlib as mpl

from sklearn.manifold import MDS

MDS()

# конвертуємо дві компоненти, оскільки точки відображаються на двовимірному просторі
# "precomputed" - оскільки надається матриця відстаней
# random_stateє - щоб графік був відтворюваним.
mds = MDS(n_components=2, dissimilarity="precomputed", random_state=1)

pos = mds.fit_transform(dist)  # форма (n_components, n_samples)

xs, ys = pos[:, 0], pos[:, 1]
print()
print()


# налаштовуємо кольори для кластерів за допомогою словника
cluster_colors = {0: '#1b9e77', 1: '#d95f02', 2: '#7570b3', 3: '#e7298a', 4: '#66a61e'}

# налаштовуємо назви кластерів за допомогою словника
cluster_names = {0: 'Family, home, war',
                 1: 'Police, killed, murders',
                 2: 'Father, New York, brothers',
                 3: 'Dance, singing, love',
                 4: 'Killed, soldiers, captain'}

# налаштування ipython для відображення графіків matplotlib
%matplotlib inline

# створюємо фрейм даних, який має результат MDS, а також номери кластерів і назви фільмів
df = pd.DataFrame(dict(x=xs, y=ys, label=clusters, title=titles))

# групуємо за кластерами
groups = df.groupby('label')
```

```
# налаштовуємо графік
fig, ax = plt.subplots(figsize=(17, 9)) # задаємо розмір
ax.margins(0.05) # додає 5% заповнення до автомасштабування

# ітеруємо через групи для накладання графіка
# використовуються словники cluster_name та cluster_color викликом
# 'name' для отримання відповідного кольору/позначення
for name, group in groups:
    ax.plot(group.x, group.y, marker='o', linestyle='', ms=12,
            label=cluster_names[name], color=cluster_colors[name],
            mec='none')
    ax.set_aspect('auto')
    ax.tick_params(\
        axis= 'x',          # зміни застосовуються до осі x
        which='both',       # використовуються обидві основні і додаткові мітки
        bottom='off',       # мітки вздовж нижнього краю вимкнені
        top='off',          # мітки вздовж верхнього краю вимкнені
        labelbottom='off')
    ax.tick_params(\
        axis= 'y',          # зміни застосовуються до осі y
        which='both',       # використовуються обидві основні і додаткові мітки
        left='off',         # мітки вздовж лівого краю вимкнені
        top='off',          # мітки вздовж верхнього краю вимкнені
        labelleft='off')

ax.legend(numpoints=1)  # відображення легенди з лише 1 точкою
# додаємо мітку в позиції x, y із позначенням y якості назви фільму
for i in range(len(df)):
    ax.text(df.ix[i]['x'], df.ix[i]['y'], df.ix[i]['title'], size=8)


plt.show() # відображення графіку

# при необхідності можна зберегти графік
# plt.savefig('clusters_small_noaxes.png', dpi=200)
```