Міністерство освіти і науки України

Національний технічний університет

«Дніпровська політехніка»

Факультет інформаційних технологій

(факультет)

Кафедра системного аналізу та управління

(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи ступеня бакалавра

Студента Рублевського Івана Олександровича

академічної групи 124-19-1

спеціальності 124 Системний аналіз

на тему: «Розробка математичної моделі оптимізації пошуку інформації в мережі інтернет з використанням методів кластеризації»

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		Рейтинговою	Інституційною	
Кваліфікаційної роботи	*Одновол М.М.*			
розділів:				
Інформаційно-аналітичний	*Одновол М.М.*			
Спеціальний розділ	*Одновол М.М.*			
Рецензент				
Нормоконтролер	*Хом`як Т.В.*			

Дніпро

2023

ЗАТВЕРДЖЕНО:

завідувач кафедри

Системного аналізу та управління
(повна назва)
_____ к.т.н., доц. Желдак Т.А.
(підпис) (прізвище, ініціали)

«____» _____2023року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня *бакалавра*

студенту *Рублевському І. О.* **академічної групи** *124-19-1*

спеціальності: *124 Системний аналіз*

на тему *«*Розробка математичної моделі оптимізації пошуку інформації в мережі інтернет з використанням методів кластеризації*»*

затверджену наказом ректора НТУ «Дніпровська політехніка»

від *18.05.2023 р. №350-с*

Розділ	Зміст	Терміни виконання
Інформаційно-аналітичний розділ	Системний аналіз математичної моделі інформаційного пошуку і постановка задачі	
Спеціальний розділ	Розробка математичної моделі оптимізації пошуку інформації з використанням методів кластеризації	

Завдання видано _____ *Одновол М.М.*
(підпис керівника) (прізвище, ініціали)

Дата видачі: _____

Дата подання до екзаменаційної комісії: _____

Прийнято до виконання _____ *Рублевський І.О.*
(підпис студента) (прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 90 с., 31 малюнків, 1 таблиця, 3 додатки, 22 джерела.

Об’єкт досліджень – процес пошуку текстових документів та інформації в мережі Інтернет.

Предмет досліджень - методи кластерізації, що використовуються для організації процесу пошуку текстових документів в мережі Інтернет та їх ефективність.

Мета кваліфікаційної роботи – оптимізація процесу пошуку текстових документів в Інтернеті шляхом реалізації математичних моделей, які базуються на методах кластеризації.

В інформаційно-аналітичному розділі розглянуто актуальні методи кластерного аналізу та принципи їх роботи. Описано принципи роботи інформаційно пошукових систем

У спеціальному розділі розглянуто два аспекти: реалізація методів у програмному середовищі та оцінка ефекиивності обраних методів у контексті інформаційного пошуку.

Практична цінність результатів полягає у піборі оптимальних методів кластеризації, спрямованих на покращення пошуку текстових документів в Інтернеті. Завдяки використанню передових математичних моделей успішно розроблена стратегії для оптимізації пошуку пов’язаних з комп’ютером текстових документів у просторі Інтернету.

КЛАСТЕР, КЛАСТЕРІЗАЦІЯ, ІНФОРМАЦІЙНИЙ ПОШУК, ІНФОРМАЦІЙНО-ПОШУКОВА СИСТЕМА, ІНТЕРНЕТ, ОПТИМІЗАЦІЯ, ЦЕНТРОЇД, ЕВКЛІДОВА ВІДСТАНЬ.

THE ABSTRACT

Explanatory note: 90 p., 31 figures, 1 table, 3 appendices, 22 sources.

The object of research is the process of searching for text documents and information on the Internet.

The subject of research is the application of clustering methods for organizing the process of searching for text documents on computer topics on the Internet and evaluating their effectiveness.

The purpose of the qualification work is to optimize the process of searching for text documents on the Internet by implementing mathematical models based on clustering methods.

In the information-analytical section, the current methods of cluster analysis and the principles of their work are considered. The principles of operation of information retrieval systems are described.

In the special section, two aspects will be considered: the implementation of methods in the software environment and the evaluation of the effectiveness of the selected methods in the context of information retrieval.

The practical value of the results lies in the selection of optimal

clustering methods aimed at improving the search for text documents on the Internet. Thanks to the use of advanced mathematical models, strategies have been successfully developed to optimize the search for computer-related text documents on the Internet.

CLUSTER, CLUSTERING, INFORMATION RETRIEVAL, INFORMATION RETRIEVAL SYSTEM, INTERNET, OPTIMIZATION, CENTROID, EUCLIDEAN DISTANCE.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ 6
ВСТУП 7
1 ІНФОРМАЦІЙНО-АНАЛІТИЧНИЙ РОЗДІЛ 11
1.1 Аналіз предметної області 11
1.1.1 Пошук інформації як об’єкт аналізу 11
1.1.2 Аналіз структури пошуку інформації в мережі Інтернет 17
1.2 Класифікація методів кластеризації 20
1.3Аналіз можливостей математичної моделі оптимізації пошуку інформації в мережі Інтернет з використанням методів кластеризації 24
1.4 Постановка завдання дослідження 29
2 СПЕЦІАЛЬНИЙ РОЗДІЛ 30
2.1 Аналіз застосовності методів кластеризації для оптимізації пошуку документів в умовах інформаційно-пошукових систем 30
2.1.1 K-means (k-середніх) 33
2.1.2 Hierarchical Clustering (ієрархічна кластеризація) 36
2.1.3 Latent Dirichlet Allocation (LDA) 39
2.1.4 DBSCAN (Density-Based Spatial Clustering of Applications with Noise) 42
2.1.5 Mean Shift 44
2.2 Вибір методів оптимізації пошуку документів в мережі Інтернет з використанням методів кластеризації 46
2.3 Кластеризація множин з використанням алгоритму K-means 50
2.4 Побудова математичної моделі кластерізації з використанням алгоритму K-means 52
2.5 Кластеризація множин з використанням алгоритму Latent Dirichlet Allocation 54
2.6 Побудова математичної моделі кластерізації з використанням алгоритму Latent Dirichlet Allocation 58
2.7 Оцінка ефективності алгоритмів Latent Dirichlet Allocation і K-means у кластеризації колекцій текстових документів 64
2.8 Оптимізація пошуку документів в мережі Інтернет з використанням алгоритмів кластеризації 77
ВИСНОВКИ 81
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ 82
ДОДАТКИ 84

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

WWW - World Wide Web. Всесвітня павутина, глобальна гіпертекстова система, використовуюча Internet в якості транспортного засобу.

LDA – Latent Dirichlet Allocation ймовірнісний генеративний алгоритм для тематичного моделювання тексту.

ІП - інформаційний пошук. Процес відшукування в деякій безлічі текстів (документів) усіх таких, які присвячені вказаній в запиті темі (предмету) або містять потрібні споживачеві факти, відомості.

ІПС - інформаційно - пошукова система. Система, що забезпечує пошук і відбір необхідних даних в спеціальній базі з описами джерел інформації (індексі) на основі інформаційно-пошукової мови і відповідних правил пошуку

ВСТУП

Актуальність роботи. У сучасних умовах, де використання природних і людських ресурсів, матеріальних і фінансових засобів стає все важливішим, особливу увагу варто приділити пошуку оптимальних рішень для різних проблем. Одним з таких завдань є оптимізація процесу пошуку текстових документів на певну тематику в мережі Інтернет. В сучасному суспільстві електронна інформація набуває все більшої значущості в усіх сферах життя. По всьому світу існують розподілені інформаційні сховища, які містять терабайти текстових даних.

Розвиток інформаційних ресурсів Інтернету призвів до серйозної проблеми інформаційного перевантаження. На сьогоднішній день ця проблема стоїть дуже гостро, оскільки обсяги баз знань постійно зростають не лише в мережі Інтернет, але й на кожному підприємстві. Головна проблема, пов'язана з інформаційним пошуком, полягає в досягненні високої точності та швидкості процесу пошуку. Кожне завдання інформаційного пошуку вимагає знаходження оптимального рішення для досягнення поставленої мети.

Аналіз предметної області в цьому напрямку підтвердив доцільність використання кластеризації для оптимізації процесу інформаційного пошуку. Застосування кластеризації дозволяє розподілити документи в колекції на класи, що сприяє покращенню швидкості пошуку і точності результатів. Однак, через велику кількість наявних методів кластеризації, була необхідність провести аналіз існуючих алгоритмів кластеризації та визначити найбільш ефективні для кластеризації колекції текстів на тему комп'ютерів. Вищезгадані проблеми підкреслюють актуальність оцінки ефективності методів кластеризації та їх використання для розробки математичних моделей, спрямованих на оптимізацію пошуку документів в мережі Інтернет.

Метою дослідження є оптимізація процесу пошуку текстових документів в мережі Інтернет та розробка математичних моделей для покращення цього процесу. Завданнями дослідження є аналіз ефективності методів кластеризації,

які застосовуються для організації пошуку текстових документів на тему комп'ютерів в мережі Інтернет.

Об'єктом дослідження є процес інформаційного пошуку текстових документів в мережі Інтернет.

Предметом дослідження є ефективність методів кластеризації, які використовуються для організації пошуку текстових документів з тематикою, пов'язаною з комп'ютерами, в мережі Інтернет.

Ідея дослідження полягає в оцінці ефективності методів кластеризації та їх використанні для створення математичних моделей пошуку текстових документів з тематикою, пов'язаною з комп'ютерами, в мережі Інтернет. Розроблені математичні моделі планується впровадити на веб-сайтах, які мають відношення до комп'ютерів, для оптимізації процесу пошуку необхідної інформації на них.

Для досягнення поставлених цілей та вирішення проблем використовувалися такі методи дослідження: аналіз та наукове узагальнення літературних джерел, які стосуються початкових засад досліджень, а також методологія кластеризації даних.

Нукові положення, очікувані наукові результати роботи передбачають оцінку ефективності методів кластеризації та їх використання для розробки математичних моделей, спрямованих на оптимізацію пошуку текстових документів на тематику, пов'язану з комп'ютерами, в мережі Інтернет. Очікувані наукові результати включають розробку математичних моделей, які проявлятимуть високу ефективність при кластеризації колекцій текстових документів з тематики, пов'язаної з комп'ютерами.

Ці наукові результати мають допомогти покращити процес пошуку необхідної інформації в мережі Інтернет, зокрема на веб-сайтах, що стосуються комп'ютерної тематики. Застосування розроблених математичних моделей сприятиме підвищенню ефективності кластеризації колекцій текстових документів та забезпеченню більш точного та швидкого пошуку необхідної інформації.

Обгрунтованість і достовірність наукових положень, висновків і рекомендацій магістерської роботи базується на коректності поставлених проблем і прийнятих допущень для побудови математичних моделей. Наукова новизна отриманих результатів проявляється у наступних аспектах:

1. Досліджено методи кластеризації і виявлено найбільш оптимальні для розробки математичних моделей оптимізації пошуку текстових документів на тематику, пов'язану з комп'ютерами, в мережі Інтернет. Це вказує на розширення знань у галузі кластерного аналізу та його застосування в контексті пошуку інформації в Інтернеті.
2. Розроблені математичні моделі оптимізації пошуку текстових документів в мережі Інтернет. Це свідчить про творчий підхід до розв'язання проблеми та створення нових методологій, які можуть сприяти покращенню ефективності пошуку інформації.
3. Розроблено методику, що містить план експериментів для оцінки ефективності і побудови математичної моделі для оцінки якості кластеризації при дослідженні ефективності алгоритмів автоматичної кластеризації текстових документів на тематику, пов'язану з комп'ютерами. Це дозволяє забезпечити науковий підхід до експериментального дослідження та об'єктивну оцінку ефективності методів.

Усі ці аспекти спільно сприяють науковій новизні отриманих результатів, підвищують обгрунтованість і достовірність висновків та рекомендацій, що викладені в кваліфікаційній роботі бакалавра.

Практичне значення отриманих результатів дослідження полягає в наступному:

1. Визначення найбільш ефективних методів кластеризації для розробки математичних моделей оптимізації пошуку текстових документів на тематику, пов'язану з комп'ютерами, в мережі Інтернет. Це дозволяє практикам і веб-розробникам обрати оптимальні методи для покращення процесу пошуку інформації в мережі.

2. Розробка ефективних математичних моделей оптимізації пошуку текстових документів на околокомпьютерну тематику в мережі Інтернет. Ці моделі можуть бути використані для покращення швидкості та точності пошуку документів на веб-сайтах, що стосуються комп'ютерів.
3. Розробка методики, яка включає план експериментів для оцінки ефективності алгоритмів автоматичної кластеризації масиву текстових документів на околокомп'ютерну тематику. Ця методика може бути використана дослідниками і практиками для оцінки якості різних алгоритмів кластеризації і вибору найбільш ефективних для конкретних завдань.

1 ІНФОРМАЦІЙНО-АНАЛІТИЧНИЙ РОЗДІЛ

1.1 Аналіз предметної області

1.1.1 Пошук інформації як об’єкт аналізу

Інформаційний пошук є процесом знаходження текстових документів, які відповідають певній темі або містять необхідну інформацію. Він може бути здійснений вручну або з використанням інформаційно-пошукової системи (ІПС) і може бути підтриманий засобами механізації або автоматизації. Термін "інформаційний пошук" був уперше введений Кельвіном Муром в 1948 в його докторській дисертації, опублікований і вживається в літературі з 1950. Спочатку системи автоматизованого ІП, або інформаційно-пошукові системи, використовувалися лише для управління інформаційним вибухом в науковій літературі. Багато університетів і публічні бібліотеки стали використовувати ІПС для забезпечення доступу до книг, журналів і інших документів. Широкого поширення ІПС набули з появою мережі Інтернет.

Найбільш популярними пошуковими системами в Україні є Google та Yandex. За даними Similarweb, на квітень 2023 року, Google займає перше місце з часткою ринку пошуку серед українців 91,02%, а Яндекс - друге місце з 5,92% Пошук інформації є процесом знаходження документів (текстів) тих, які стосуються вказаної теми (предмету) і задовольняють певні умови пошуку (запиту), або містять необхідні факти, відомості або дані, що відповідають інформаційним потребам користувача. Схема процесу пошуку представлена на рисунку 1.1.

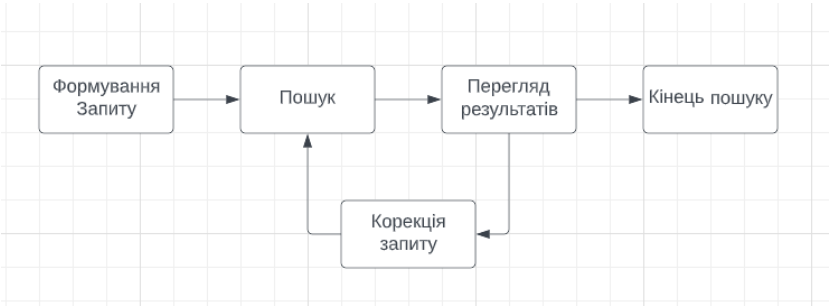


Рисунок 1.1 - Процес пошуку

ІП є процесом знаходження та витягування існуючої інформації з великої кількості документів або джерел. Він зазвичай базується на заданому запиті користувача та використовує інформаційно-пошукову систему (ІПС) для знаходження та представлення відповідних даних.

У контексті ІП, факти або відомості, що можуть бути знайдені, обмежуються тим, що були введені в систему або індексовані нею. ІПС збирає, індексує та зберігає інформацію, а потім використовує цей індекс для швидкого знаходження відповідної інформації під час пошуку. Таким чином, ІПС може забезпечити доступ до фактів або відомостей, які були введені в систему, але не здатна відповісти на запити, які виходять за межі наявної інформації.Логічна переробка інформації, з іншого боку, передбачає аналіз та інтерпретацію інформації, застосування розумових процесів, аргументацію та роботу зі знаннями для формулювання відповідей на запити. Це вимагає більш складних алгоритмів та інтелектуальних можливостей, що виходять за рамки простого пошуку інформації.

Перед введенням тексту або документу в інформаційно-пошукову систему (ІПС), спочатку визначається його основний смисловий зміст, тема або предмет. Потім цей смисловий зміст перекладається і записується на одній з інформаційно-пошукових мов у вигляді пошукового чину тексту. Такий же підхід використовується при введенні фактів або відомостей в ІПС. Запит, який надійшов до системи, також перекладається інформаційно-пошуковою мовою, утворюючи пошуковий припис.

Оскільки пошукові образи текстів і пошукові приписи записані на одній і тій же мові, вирази на цій мові можуть мати тільки одне тлумачення. Тому можна формально порівнювати їх без необхідності розуміти їх семантику. Для цього встановлюються певні правила або критерії відповідності, які вказують, в якій мірі формальний збіг між пошуковим образом та пошуковим приписом тексту повинен бути вважаний достатнім для відповіді на інформаційний запит і надання відповідного результату.

Розглянемо типову схему інформаційно-пошукової системи на рисунку 1.2 .

Client - це програма, призначена для перегляду конкретних інформаційних ресурсів. На сьогоднішній день, популярними є мультипротокольні програми, такі як веб-браузери типу Netscape Navigator. Ці програми дозволяють переглядати документи на World Wide Web, Gopher, Wais, FTP-архівах, поштових списків розсилки та груп новин Usenet. У свою чергу, всі ці інформаційні ресурси є об'єктом пошуку в інформаційно-пошуковій системі.

Інтерфейс користувача (User Interface) в інформаційно-пошуковій системі відноситься не лише до програми перегляду, але і до способу взаємодії користувача з пошуковими функціями системи, включаючи формування запитів і перегляд результатів пошуку. Важливо розрізняти між процесом перегляду результатів пошуку і самими інформаційними ресурсами у мережі, оскільки це дві різні сутності, про які ми можемо говорити окремо. Детальніше на цьому можна зупинитися пізніше.

Пошукова машина (Search engine) виконує функцію перетворення запиту користувача, який формулюється на інформаційно-пошуковій мові, в формальний запит системи, що включає пошук посилань на інформаційні ресурси в Мережі та видачу результатів пошуку користувачеві. Робота пошукової машини базується на індексі, який містить інформацію про доступні ресурси в Мережі. Після отримання запиту від користувача, пошукова машина виконує пошук у своїй базі даних та індексі, і видає результати, що відповідають поставленому запиту.

Index database - індекс - це основний масив даних інформаційно-пошукової системи. Він служить для пошуку адреси інформаційного ресурсу. Архітектура індексу влаштована так, щоб пошук відбувався максимально швидко і при цьому можна було б оцінити цінність кожного зі знайдених інформаційних ресурсів мережі.

Queries - запити користувача зберігаються в його особистій базі даних. На відладку кожного запиту йде досить багато часу, і тому надзвичайно важливо зберігати запити, на які система дає хороші відповіді.

Index robot - Робот-індексатор (іноді відомий як "павук" або "пошуковий робот") - це програма, яка виконує сканування Інтернету для збору інформації про веб-сторінки і підтримує базу даних індексу в актуальному стані. Цей процес відомий як індексація. Робот-індексатор використовується пошуковими системами для автоматичного збору даних про веб-сторінки та їх вміст. Він переходить по посиланнях, що зустрічаються на сторінках, і збирає інформацію про ці сторінки, таку як URL, заголовок, мета-теги, текстовий контент та інші важливі деталі.

Www sites - це увесь Internet. А якщо говорити точніше, то це ті інформаційні ресурси, перегляд яких забезпечується програмами перегляду.

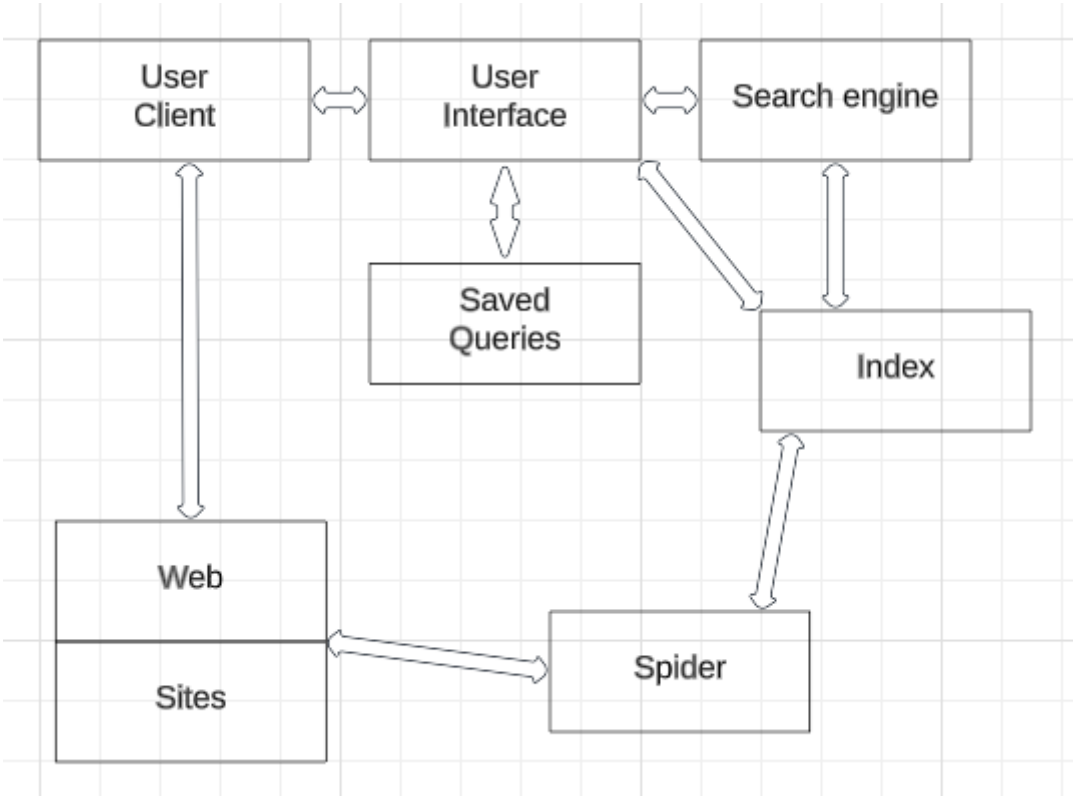


Рисунок 1.2 - Структура ІПС в мережі Інтернет

Говорячи про інформаційно-пошукові системи, терміни "запит" і "об'єкт запиту" використовуються для опису формалізованого способу вираження інформаці Одним із важливих вимог користувачів щодо інформаційного пошуку є можливість отримати поверхневе уявлення про список знайдених документів, не обов'язково переглядаючи весь вміст цих документів. Це дозволяє швидко оцінити відповідність результатів пошуку поставленому запиту та визначити, які документи можуть бути найбільш корисними або цікавими.

Проблеми, пов'язані з ефективністю інформаційного пошуку, зацікавлюють не лише фахівців, але й широке коло дослідників. Кількість наукових робіт, присвячених покращенню ефективності пошуку інформації в Інтернеті, постійно зростає з кожним роком. Це свідчить про значимість проблеми та постійне прагнення до вдосконалення алгоритмів, технологій та підходів до пошуку та представлення результатів користувачам. Запит представляє собою вимогу або питання, які користувач ставить перед системою з метою отримання відповідної інформації.

Для вираження інформаційної потреби використовується мова пошукових запитів, яка може мати різний синтаксис в залежності від конкретної інформаційно-пошукової системи. Цей синтаксис визначає правила і формати, за якими користувач формулює запити для пошуку конкретної інформації.

Окрім спеціальної мови запитів, сучасні пошукові системи дозволяють також вводити запити на природній мові. Це означає, що користувач може використовувати звичайну мову, таку як англійська, для формулювання запиту без необхідності вивчати спеціальний синтаксис пошукових запитів.

Об'єкт запиту в інформаційно-пошуковій системі відображає сутність інформації, яка зберігається в базі даних системи. Хоча найпоширенішим об'єктом запиту є текстові документи, принципових обмежень немає. В залежності від системи, можливий пошук інших типів інформації, таких як зображення, музика, відео тощо. Об'єкт запиту визначає те, що саме користувач хоче знайти або отримати інформацію про це.

Процес занесення об'єктів пошуку в інформаційно-пошукову систему називається індексацією. Далеко не завжди ІПС зберігає точну копію об'єкту, нерідко замість неї зберігається сурогат.

Технічна ефективність інформаційної пошукової системи (ІПС) може бути оцінена за допомогою двох відносних показників - коефіцієнта точності і коефіцієнта повноти.

1. Коефіцієнт точності відображає, наскільки точно система відповідає на інформаційний запит. Чим більший коефіцієнт точності, тим більш точні результати пошуку надає система.
2. Коефіцієнт повноти вказує на те, яку частину всіх доступних текстів, що відповідають на запит, система здатна знайти. Високий коефіцієнт повноти свідчить про те, що система знаходить значну частину інформації, що відповідає на запит.

Ці два показники допомагають оцінити якість і ефективність ІПС з точки зору точності і повноти пошуку. Максимальна ефективність досягається, коли коефіцієнт точності і коефіцієнт повноти максимально високі. значення коефіцієнта точності і коефіцієнта повноти можуть змінюватися в залежності від специфіки інформаційних потреб та конкретного використання інформаційної пошукової системи.

У випадку пошуку патентних описів з метою проведення експертизи патентної заявки на новизну, важливо досягти максимальної повноти видачі, тобто знайти всі належні патентні описи, що відповідають запиту. Тому в цьому випадку 100% -на повнота є необхідною.

У випадку пошуку, орієнтованого на звичайного дослідника або інженера, вважається, що дуже хорошою точністю видачі є значення близько 80%. Це означає, що більшість результатів пошуку будуть точними і відповідатимуть на запит.

Зазначені значення є лише прикладами і можуть варіюватися в залежності від конкретних потреб і вимог користувача.

Центральна мета інформаційно-пошукової системи (ІПС) полягає у наданні користувачеві задоволення його інформаційних потреб. Запит відображає інформаційну потребу користувача та допомагає інформаційно-пошуковій системі зрозуміти, які ресурси або документи можуть відповідати цим потребам. Ключові слова в запиті використовуються для ідентифікації та вибору відповідних інформаційних ресурсів, які найкраще відповідають запиту користувача. Запит є засобом комунікації між користувачем і системою, допомагаючи зрозуміти інформаційні потреби користувача і забезпечуючи релевантні результати пошуку.

Одним з основних завдань, з яким почався розвиток інформаційно-пошукових систем, було пошук документів, що задовольняють запит користувача, у рамках певної статичної колекції документів. Але список завдань ІП постійно розширюється і тепер включає:

- Питання моделювання;
- Класифікація документів;
- Фільтрація документів;
- Кластеризація документів;
- Проектування архітектури пошукових систем і призначених для користувача інтерфейсів;
- Витягання інформації, зокрема анотування і реферування документів;
- Мови запитів та ін.

1.1.2 Аналіз структури пошуку інформації в мережі Інтернет

Однією з ключових проблем інформаційного пошуку є потреба в ефективній обробці значних обсягів текстової інформації. Ця проблема виникає через велику кількість документів, які потрібно проаналізувати під час обробки запиту користувача. Внаслідок цього, тривалість обробки запиту може бути значною, що не задовольняє сучасних користувачів, які очікують швидких і точних результатів.

Одним із важливих вимог користувачів щодо інформаційного пошуку є можливість отримати поверхневе уявлення про список знайдених документів, не обов'язково переглядаючи весь вміст цих документів. Це дозволяє швидко оцінити відповідність результатів пошуку поставленому запиту та визначити, які документи можуть бути найбільш корисними або цікавими.

Проблеми, пов'язані з ефективністю інформаційного пошуку, зацікавлюють не лише фахівців, але й широке коло дослідників. Кількість наукових робіт, присвячених покращенню ефективності пошуку інформації в Інтернеті, постійно зростає з кожним роком. Це свідчить про значимість проблеми та постійне прагнення до вдосконалення алгоритмів, технологій та підходів до пошуку та представлення результатів користувачам.

Перед аналізом методів вирішення проблем інформаційного пошуку, важливо згадати про найчастіше виникаючі завдання в цій області. Класифікація і кластеризація є двома такими завданнями, які часто вирішуються при пошуку інформації, особливо в контексті текстових документів.

1. Класифікація: Класифікація полягає в призначенні категорії або мітки до документів залежно від їх вмісту. Це дозволяє організувати документи у структуровану систему, де вони розподіляються за певними критеріями. У роботі [1] згадується, що класифікація відносить кожен об'єкт до однієї із заздалегідь визначених груп, які не завжди можуть бути відомі при організації пошуку. Класифікація може бути заснована на попередньо визначених категоріях або може вимагати автоматичного виявлення

тематики або характеристик документів. У роботі [2] автор згадує і доводить те, що при інформаційному пошуку кількість об'єктів і їх атрибутів може бути дуже великою і ефективність класифікації буде низькою, тому мають бути передбачені інтелектуальні механізми оптимізації процесу класифікації або задействованны інші методи угрупування текстових масивів.

2. Кластеризація: Кластеризація дійсно подібна до класифікації, але має свої особливості. Вона є логічним продовженням класифікації, але відрізняється тим, що класи або кластери об'єктів у вивчуваному наборі даних не задаються заздалегідь. Терміни "кластеризація", "автоматична класифікація", "навчання без учителя" і "таксономія" є синонімами.

   Кластеризація спрямована на розбиття сукупності об'єктів на однорідні групи, які називаються кластерами або класами. Якщо представити ці об'єкти як точки в просторі ознак, то завдання кластеризації полягає у визначенні "згущень точок". Основна мета кластеризації - знайти наявні структури в даних. Кластеризація є описовою процедурою, що дозволяє проводити розвідувальний аналіз і вивчати "структуру даних". Вона не здійснює статистичних висновків, але допомагає виявити патерни, залежності та групування об'єктів у великому наборі даних.

   Кластеризація є потужним інструментом для виявлення структури і організації даних, дозволяє здійснювати експлоративний аналіз і відкривати нові знання. Вона застосовується у багатьох галузях, включаючи машинне навчання, аналіз тексту, соціальні мережі, маркетингові дослідження та багато інших.

   .

Ці завдання є важливими при обробці текстових документів, оскільки дозволяють організувати інформацію, зрозуміти зв'язки між документами і полегшити пошук інформації у великих колекціях документів. Методи класифікації і кластеризації можуть використовуватись як окремо, так і в

поєднанні з іншими методами інформаційного пошуку для поліпшення точності та ефективності результатів пошуку.

Кластеризація сьогодні знаходить широке застосування в різних сферах для різних цілей. Одним із важливих використань кластеризації є реферування великих документальних масивів. Шляхом кластеризації документів можна визначити взаємозв'язані групи документів, що допомагає спростити процес перегляду та пошуку необхідної інформації. За допомогою кластеризації також можна виявити унікальні документи з колекції, що можуть бути цікавими для подальшого дослідження або аналізу. Крім того, кластеризація може допомогти виявити дублікати або дуже близькі за змістом документи, що є важливим при обробці великих обсягів інформації.

Сьогодні кластеризація текстових документів, завдяки розподілу документів в колекції по класах, є однією з найважливіших та динамічно розвиваючихся областей інформаційних технологій, оскільки вона дозволяє покращити швидкість пошуку документів та точність наданої відповіді. Кластеризація як фундаментальний метод використовується в багатьох областях, таких, як data mining [1], [2], інформаційного пошуку [3-5], виявлення тематики [6] і так далі. У зоні інформаційного пошуку для поліпшення ефективності широко застосовується кластеризація документів. Кластеризація, тобто розбиття сукупності документів на кластери схожих по контенту, є одним з методів підготовки документів до пошуку

Якщо прямий пошук не дає бажаних результатів, то кластеризована колекція документів може бути прийнята як результат пошуку. Кластеризація надає можливість організувати документи у групи або кластери на основі їхньої схожості. Кожен кластер може представляти певну тему або концепцію, що допомагає користувачеві зорієнтуватись у колекції документів.

1.2 Класифікація методів кластеризації

Зараз ми розглянемо класифікацію методів кластерного аналізу. Методи кластерного аналізу можна розділити на дві групи:

1. Ієрархічні методи кластерного аналізу:
    - Агломеративний метод: цей метод починає з кожного об'єкта в окремому кластері і послідовно об'єднує найближчі кластери до тих пір, поки не отримається один загальний кластер. В результаті отримується деревоподібна структура, відображаюча ієрархію кластерів.
    - Дивізивний метод: цей метод починає з одного великого кластера, який поступово розбивається на менші кластери шляхом ділення на підгрупи. Процес розбиття продовжується досягнення бажаної кількості кластерів або заданої умови розбиття.
2. Неієрархічні методи кластерного аналізу:
    - K-means: цей метод визначає кластери шляхом розділення об'єктів у просторі признаків на задану кількість кластерів. Він базується на знаходженні центроїдів кластерів і призначенні кожного об'єкта до найближчого центроїда.
    - DBSCAN: цей метод використовує густоту точок у просторі признаків для виявлення кластерів. Він визначає об'єкти, які знаходяться у густих областях як кластери, і виявляє шумові об'єкти або об'єкти, що не входять до кластерів.
    - Статистичні методи: вони використовують статистичні моделі і алгоритми для кластеризації даних. Наприклад, метод гаусівської суміші (Gaussian Mixture Model) визначає кластери, розподіляючи дані відповідно до гаусівських розподілів.

Суть ієрархічної кластеризації полягає в послідовному об'єднанні менших кластерів у більші або поділі великих кластерів на менші. Ієрархічні методи поділяються на агломеративні та дивізіональні.

Перевагою агломеративних методів є їх простота і можливість візуалізації ієрархічної структури. Однак, вони можуть бути обчислювально витратними для великих наборів даних через необхідність обчислення відстаней між кластерами на кожному кроці.

Агломеративні методи є однією з підгруп ієрархічних методів кластерного аналізу. Ці методи починають з розгляду кожного об’єкта як окремого кластера і поступово об’єднують найближчі кластери досягнення одного загального кластера, який містить всі об’єкти [6]. Основна ідея агломеративних методів полягає в послідовному об’єднанні найближчих кластерів, що приводить до формування деревоподібної структури, відображаючої ієрархію кластерів [7].

Один з найпоширеніших агломеративних методів є метод “Complete Linkage” (повне зв’язування), який визначає відстань між кластерами як максимальну відстань між їх об’єктами. Інші варіанти агломеративних методів використовують інші критерії для об’єднання кластерів, такі як середня відстань або мінімальна відстань [7] .

Перевагою агломеративних методів є їх простота і можливість візуалізації ієрархічної структури. Однак, вони можуть бути обчислювально витратними для великих наборів даних через необхідність обчислення відстаней між кластерами на кожному кроці [7].

Агломеративні методи кластерного аналізу включають такі підходи:

1. Метод ближнього сусіда (поодинокий зв'язок) - відстань між кластерами визначається відстанню між двома найближчими об'єктами в різних кластерах.
2. Метод найбільш видалених сусідів (повний зв'язок) - відстань між кластерами визначається найбільшою відстанню між будь-якими двома об'єктами в різних кластерах.
3. Метод незваженого попарного середнього - відстань між кластерами обчислюється як середня відстань між усіма парами об'єктів у кластерах.
4. Метод зваженого попарного середнього - схожий на метод незваженого попарного середнього, але використовується ваговий коефіцієнт, яким є розмір кластера.

Ці методи дозволяють утворити кластери залежно від відстаней між об'єктами в колекції.

Дисперсійні методи агломеративної кластеризації використовують міру внутрішньокластерної дисперсії для об'єднання кластерів. Ці методи базуються на ідеї, що об'єкти в межах одного кластера повинні бути схожі між собою, а об'єкти з різних кластерів - відмінні. Нижче наведено декілька популярних дисперсійних методів:

1. Метод середньої абсолютної відхиленості (average linkage) - він використовує міру відстані між середніми значеннями об'єктів у кластерах для об'єднання. Міра відстані може бути обчислена за допомогою різних метрик, наприклад, Евклідової відстані.
2. Метод Wards (Ward's linkage) - цей метод мінімізує збільшення внутрішньокластерної суми квадратів відхилень при об'єднанні кластерів. Він використовується для збалансованої кластеризації і зазвичай дає компактні кластери.
3. Метод центроїдів (centroid linkage) - використовується середнє арифметичне або медіана координат об'єктів у кластерах як центроїд для обчислення відстані між кластерами.

Кожен з цих методів має свої переваги та особливості. Вибір конкретного дисперсійного методу залежить від природи даних та постановки задачі кластеризації текстових документів[8].

До центроїдних методів агломеративної кластеризації належать:

1. Метод середнього посилання (average linkage): Він використовує середнє значення відстаней між всіма парами об'єктів у різних кластерах для об'єднання. Цей метод враховує всі пари об'єктів при обчисленні відстаней і дозволяє створювати багатоформатні кластери.
2. Метод медіанного зв'язку (median linkage): Він використовує медіанне значення відстаней між всіма парами об'єктів у різних кластерах для об'єднання. Цей метод є менш чутливим до викидів (аномалій) в даних, оскільки використовує медіану замість середнього значення.
3. Метод центроїдів (centroid linkage): Він використовує центроїди кластерів, що представляють собою середнє значення або середнє арифметичне значення координат об'єктів у кластері, для обчислення відстані між кластерами. Цей метод створює компактні кластери, оскільки використовує центроїди для об'єднання.

Ці методи базуються на використанні центральних точок (центроїдів) для вимірювання відстаней між кластерами[7].

Ієрархічні ділимі методи (DIvisive ANAlysis). Ці методи працюють у протилежний спосіб до агломеративних методів. Починаючи з одного великого кластеру, на кожному кроці алгоритм розбиває цей кластер на менші підкластери. Цей процес продовжується досягнення потрібного рівня деталізації або до відповідного критерію зупинки.

На початку роботи алгоритму усі об'єкти належать до одного кластеру, і потім кожен кластер поступово розщеплюється на менші кластери. Цей процес створює послідовність розщеплюючих груп, де кожна група є підкластером більшої групи.

Ієрархічні ділимі методи можуть бути корисними, коли потрібно докладніше розбити вихідний кластер на більш дрібні підкластери або коли виникає потреба в детальному аналізі структури даних. Однак, вони можуть бути обчислювально витратними, особливо при великій кількості об'єктів.

Неієрархічні методи кластерного аналізу включають:

1. Послідовний пороговий метод (Sequential Threshold Method): У цьому методі об'єкти групуються послідовно за допомогою порогового значення. Спочатку вибирається кластерний центр, а потім об'єкти, які знаходяться в межах заданого порогового значення від центру, додаються до кластеру. Цей процес повторюється для негрупованих точок, поки всі об'єкти не будуть включені до кластерів.

2. Паралельний пороговий метод (Parallel Threshold Method): У цьому методі одночасно вибираються кілька кластерних центрів, і об'єкти групуються з найближчим центром, що задовольняє пороговому рівню. Кожен кластер формується незалежно від інших кластерів, і процес повторюється для негрупованих точок.
3. Метод оптимізуючого розподілу (Optimizing Partitioning Method): Цей метод дозволяє перерозподіляти об'єкти між кластерами, щоб оптимізувати заданий критерій, наприклад, мінімізувати сумарну внутрішню відстань в кластері. В цьому методі спочатку випадковим чином вибираються кластерні центри, а потім об'єкти розподіляються до кластерів на основі їх відстаней до центрів. Процес розподілу може повторюватися, змінюючи кластерні центри і перерозподіляючи об'єкти, доки не буде досягнута оптимальна кластеризація.

Ці неієрархічні методи надають більшу гнучкість у виборі кількості кластерів та можливості оптимізувати критерії кластеризації, але вони можуть бути більш чутливими до початкового вибору кластерних центрів і залежати від порядку спостережень у даних[9].

1.3Аналіз можливостей математичної моделі оптимізації пошуку інформації в мережі Інтернет з використанням методів кластеризації

Кластерний аналіз (англ. Data clustering) - це завдання розбиття заданої вибірки об'єктів або ситуацій на підмножини, відомі як кластери, з метою сформувати кожен кластер з схожих об'єктів, при цьому об'єкти різних кластерів суттєво відрізняються між собою. Кластерний аналіз відноситься до статистичної обробки і є одним із широкого класу методів навчання без учителя. Він є багатовимірною статистичною процедурою, яка збирає дані, що містять інформацію про вибірку об'єктів, і розподіляє їх у порівняно однорідні групи, використовуючи методи, такі як Q-кластеризація або Q-техніка [10].

Кластерний аналіз дозволяє виявити приховані структури або закономірності в даних, що допомагає в розумінні внутрішніх зв'язків та характеристик груп об'єктів. Це має велике значення в багатьох областях, зокрема в пошуку інформації в мережі Інтернет.

Кластерний аналіз, походить від англійського слова "cluster" - гроно, скупчення, і знайшов своє перше застосування в соціології. Це завдання розбиття вибірки об'єктів на однорідні групи або кластери з метою виявлення структури та закономірностей у даних. Кластерний аналіз може бути застосований для групування об'єктів з різними ознаками і не обмежується конкретними типами даних.

Важливою перевагою кластерного аналізу є можливість розбиття об'єктів на основі багатьох ознак. Він також дозволяє компактно та наочно представити великі обсяги інформації, зменшуючи їх кількість. Кластерний аналіз має значення для виявлення груп тимчасових рядів, наприклад, для аналізу економічного розвитку[10].

Проте, кластерний аналіз має свої недоліки і обмеження. Вибір критеріїв розбиття впливає на кількість і склад кластерів. Під час узагальнення даних можуть втрачатися індивідуальні риси об'єктів. Також важливо враховувати можливість відсутності значень кластерів у деяких об'єктів.

Кластерний аналіз можна проводити ітеративно, враховуючи результати попередніх циклів дослідження. Цей процес може коригувати напрямок і підходи до подальшого застосування кластерного аналізу.

Таким чином, кластерний аналіз є потужним інструментом для групування та виявлення структури в даних, але його застосування потребує уважного вибору критеріїв і розуміння його обмежень.

В кластерному аналізі вважається, що:

а) Вибрані характеристики допускають в принципі бажане розбиття на кластери: При використанні кластерного аналізу важливо обрати відповідні

характеристики аналізу, які відображають схожість або відмінність між об'єктами. Якщо обрані характеристики не мають достатньої варіації або не відрізняються між об'єктами, це може призвести до неправильного розбиття на кластери. Тому важливо вибирати характеристики, які належним чином розкривають сутність об'єктів і можуть бути використані для виявлення схожості чи відмінності між ними.

б) Одиниці виміру (масштаб) вибрані правильно: При використанні кластерного аналізу важливо правильно масштабувати вибрані характеристики, оскільки неправильний масштаб може спотворити результати аналізу. Наприклад, якщо характеристики мають різні одиниці виміру або широкий діапазон значень, це може призвести до перекосу у внеску окремих характеристик у кластерний аналіз. Тому важливо проводити масштабування або нормалізацію даних перед застосуванням кластерного аналізу, щоб забезпечити правильність порівняння та оцінки схожості між об'єктами.

Цілі кластеризації включають:

1. Розуміння даних шляхом виявлення кластерної структури: Кластеризація дозволяє розбити вибірку на групи схожих об'єктів, що спрощує подальшу обробку даних і прийняття рішень. Кожен кластер може бути підданий окремому методу аналізу, що допомагає отримати глибше розуміння даних.

2. Стискування даних: Кластерний аналіз може бути використаний для скорочення великого обсягу даних. Замість використання всіх об'єктів вибірки, можна залишити по одному представнику від кожного кластера, що дозволяє зменшити обсяг даних зберігаючи при цьому важливу інформацію.

3. Виявлення новизни: Кластерний аналіз дозволяє виявити нетипові об'єкти, які не відносяться до жодного з кластерів. Це може бути корисно для виявлення аномалій або нових неочікуваних патернів, які не були враховані у вихідних кластерах.

Усі ці цілі можуть бути досягнуті за допомогою ієрархічної кластеризації, де великі кластери розбиваються на менші, які в свою чергу можуть бути дроблені ще дрібніше. Це дозволяє здійснювати більш деталізований аналіз структури даних і розрізнення більш дрібних груп об'єктів.

Кластерний аналіз виконує наступні основні завдання:

1. Розробка типології або класифікації: Кластерний аналіз допомагає створити типологію або класифікацію об'єктів на основі їх схожості. Це дозволяє групувати об'єкти в кластери з подібними характеристиками або властивостями.

2. Дослідження корисних концептуальних схем групування об'єктів: Кластерний аналіз дозволяє виявляти корисні концептуальні схеми групування

об'єктів. Це може виявити нові патерни або структури в даних, які не були відомі заздалегідь.

3. Породження гіпотез на основі дослідження даних: Кластерний аналіз може служити джерелом гіпотез про взаємозв'язки або властивості об'єктів у кожному кластері. Виявлені кластери можуть вказувати на певні закономірності або відмінності між групами об'єктів.

4. Перевірка гіпотез або дослідження для визначення присутності типів (груп): Кластерний аналіз дозволяє перевірити гіпотези про наявність певних типів або груп об'єктів у вибірці. Аналіз результатів кластеризації допомагає підтвердити або спростувати попередні припущення.

Незалежно від предмета вивчення, застосування кластерного аналізу включає наступні етапи:

1. Відбір вибірки для кластеризації: Вибірка містить об'єкти, які підлягають аналізу і групуванню.

2. Визначення безлічі змінних: Визначаються характеристики або змінні, за якими об'єкти будуть оцінюватись у вибірці.

3. Визначення метрики: Обирається метрика, що вимірює відстань або схожість між об'єктами на основі їх характеристик.

4. Застосування методу кластерного аналізу: Застосовується метод кластерного аналізу для створення груп схожих об'єктів. Це може бути ієрархічний або нерозривний підхід.

5. Представлення результатів і аналіз якості кластеризації: Результати кластеризації представляються у вигляді груп об'єктів. Якість кластеризації може бути оцінена за допомогою різних метрик і методів.

Розглянемо кожен етап детальніше.

1. Відбір вибірки: На цьому етапі обираються дані, які будуть використовуватись для кластеризації. Важливо, щоб показники не корелювали між собою, були безрозмірними, мали близький до нормального розподіл та були стійкими до впливу випадкових чинників. Вибірка повинна бути однорідною і не містити викидів.

2. Визначення змінних: На цьому етапі вибираються властивості, за якими оцінюються об'єкти вибірки. Ці змінні можуть бути якісними (наприклад, колір, статус), так і кількісними (наприклад, координати, інтервали). Для прискорення процесу кластеризації можна спробувати зменшити розмірність простору змінних і виділити найбільш важливі властивості об'єктів.

3. Вибір метрики: На цьому етапі визначається метрика, за допомогою якої буде вимірюватись близькість між об'єктами. Вибір метрики залежить від простору, в якому розташовані об'єкти, і характеристик кластерів. Деякі з поширених метрик включають Евклідову відстань, косинусну схожість, кореляцію і Хеммінгову відстань.

4. Застосування методу кластерного аналізу: На цьому етапі використовується обраний метод кластеризації для утворення груп схожих об'єктів. Деякі з поширених методів включають ієрархічну кластеризацію, k-середніх, агломеративну кластеризацію та DBSCAN.

5. Представлення результатів: На цьому етапі результати кластеризації представляються в зручному форматі для подальшого аналізу. Це може включати представлення кластерів центроїдами, характерними точками або обмеженнями, а також візуалізацію результатів на графіках або діаграмах[10].

Формальне завдання кластеризації полягає в наступному:

Дано:

- Множина об'єктів, яку потрібно розділити на групи (кластери).
- Набір атрибутів (змінних), за якими оцінюються об'єкти.

Задача:

- Знайти поділ цих об'єктів на кластери таким чином, щоб об'єкти в одному кластері були схожі між собою, а об'єкти в різних кластерах були відмінні.

Формально, це завдання можна сформулювати як оптимізаційну задачу, де метою є мінімізація (або максимізація) певної об'єктивної функції, яка відображає міру подібності (або відмінності) між об'єктами, але залежить від конкретного методу кластеризації.

Залежно від методу кластеризації, формальна постановка може додатково включати такі елементи:

- Кількість кластерів: чи відома апріорі кількість кластерів, або потрібно визначити її в процесі кластеризації.

• Тип кластерів: чи обмежений тип кластерів (наприклад, сферичні, лінійні), або кластери можуть бути будь-якої форми.

• Додаткові обмеження: можуть бути встановлені додаткові обмеження, які впливають на процес кластеризації (наприклад, мінімальний розмір кластера, обмеження на відстань між об'єктами тощо).

Або

Нехай X — множина об'єктів, Y — множина номерів (імен, міток) кластерів. Задано функцію відстані між об'єктами $\rho(x, x')$. . Є кінцева вибірка об'єктів $X^m = \{x_1, \dots, x_m\} \subset X$.. Потрібно розбити вибірку на непересічні підмножини, що називаються кластерами, так, щоб кожен кластер складався з об'єктів, близьких по метриці ρ, а об'єкти різних кластерів істотно відрізнялися. При цьому кожному об'єкту $x_i \in X^m$ приписується номер кластеру y_i.

Алгоритм кластеризації - це функція $a: X \to Y$, яка будь-якому об' єкту $x \in X$ ставити у відповідність номер кластера $y \in Y$. Множина Y в деяких випадках відома заздалегідь, проте частіше ставиться завдання визначити оптимальне число кластерів, з погляду деякого критерію якості кластеризації[10].

В кластерному аналізі, як виправдано зазначаєте, використовуються певні поняття, які важливо визначити. Основні з них:

1. Об'єкт: В контексті кластерного аналізу об'єктом може бути будь-що, що підлягає класифікації або групуванню. У вашому випадку об'єктами є документи або їх фрагменти, які представлені набором слів.

2. Термін: Терміни є словами або фразами, які використовуються для визначення особливих характеристик або ознак документів. Терміни є елементарними ознаками, і вони утворюють простір документів.

3. Простір документів: Це простір, що складається з усіх можливих комбінацій термінів, що використовуються для представлення документів. Кожен документ може бути розглянутий як вектор у цьому просторі, де кожна координата відповідає значущості певного терміна для цього документу.

4. Значущість терміна: Це величина, яка відображає важливість або частоту зустрічі терміна в документі. Значущість може бути визначена різними способами, наприклад, за допомогою частотного аналізу, вагових схем (наприклад, TF-IDF) або інших методів[5].

1.4 Постановка завдання дослідження

Величезні обсяги інформації в мережі Інтернет часто призводять до того, що кількість об'єктів, що повертаються по запиту користувача, є дуже великою. Однак у більшості випадків цю інформацію можна зробити доступною для сприйняття, якщо вміти розбивати джерела інформації на тематичні групи. Такий процес групування даних здійснюється за допомогою кластеризації.

Кластеризація дозволяє згрупувати схожі об'єкти разом, формуючи кластери або групи, які відображають спільні теми або характеристики. Це допомагає зменшити кількість об'єктів, що відображаються користувачу, і зробити інформацію більш узагальненою і легкозрозумілою. Кластеризація дозволяє структурувати інформацію, забезпечуючи логічне розподіл документів на основі їхньої схожості.

Цей процес допомагає користувачеві швидше зорієнтуватись у великій кількості даних, знаходячи групи, які цікавлять його найбільше. Кластеризація стала невід'ємною частиною процесу пошуку інформації в мережі Інтернет, допомагаючи ефективніше організувати та спростити доступ до потрібної інформації.

З метою оптимізації пошуку інформації в мережі Інтернет в кваліфікаційній роботі поставленны такі завдання для дослідження:

1. Аналіз методів кластеризації, відповідних для завдання кластеризації колекції текстових документів невеликого об'єму: K-means, DBSCAN (Density-Based Spatial Clustering of Applications with Noise), Hierarchical Clustering (ієрархічна кластеризація), Latent Dirichlet Allocation (LDA), Mean Shift
2. Визначення ефективності найбільш оптимальних методів для вирішення завдання кластеризації колекції текстових документів:

2 СПЕЦІАЛЬНИЙ РОЗДІЛ

2.1 Аналіз застосовності методів кластеризації для оптимізації пошуку документів в умовах інформаційно-пошукових систем

У сучасній класифікації і кластеризації текстових документів існує безліч методик і алгоритмів. Деякі з цих методик вже застосовуються в робочих пошукових системах. У даній дипломній роботі буде надано детальний опис і порівняння декількох відомих методів, які відрізняються за своїми алгоритмами роботи і можуть бути успішно реалізовані на практиці. Вибір цих методів здійснювався з метою включення різних підходів та отримання найкращих результатів після аналізу методів.

Для проведення дослідження набору методів, які базуються на алгоритмах кластеризації, були сформульовані критерії оцінки досліджуваних методів з точки зору кінцевого користувача системи та з точки зору реалізації методу. Ці критерії допоможуть оцінити ефективність і придатність кожного методу для практичного застосування.

З точки зору кінцевого користувача системи, що реалізовує групування документів :

1. Співвідношення швидкості і точності: Визначення оптимального балансу між швидкістю роботи алгоритму і його точністю. Деякі алгоритми можуть працювати швидше, але мають меншу точність, тоді як інші можуть бути більш точними, але потребують більше часу для обробки. Важливо вибрати метод, який дозволяє користувачеві налаштовувати це співвідношення відповідно до його потреб.
2. Можливість роботи в "інкрементному" режимі: Здатність алгоритму обробляти нові документи по мірі їх надходження без необхідності повного перерахунку всіх документів. Це важливо для систем, які отримують постійний потік даних і потребують негайного оновлення кластеризації.
3. Можливість перетину документів в рубриках: Деякі методи дозволяють одному документу належати до кількох рубрик, основаних на суміжних

темах. Це корисно, коли документи мають багатогранну природу і можуть відноситись до декількох тематичних категорій одночасно.

4. Мінімальна необхідна попередня інформація: Методи, які потребують мінімального обсягу попередньої інформації, є зручними, оскільки користувачам не потрібно витрачати багато часу і зусиль на підготовку даних або задання параметрів.
5. Обмеження для набору даних: Оцінка того, наскільки методи ефективно працюють з різними обсягами даних і враховують можливі обмеження, такі як обсяг пам'яті,

З точки зору реалізації методу виділено ще декілька критеріїв:.

1. Здатність використовувати заздалегідь обчислені характеристики: Методи, які можуть використовувати заздалегідь обчислені характеристики, такі як матриця близькості між документами(similarity matrix) або матриця вагів документів(tf·idf). Це дозволяє зберегти час і ресурси, які були витрачені на попередню обробку даних.
2. Необхідність навчання алгоритму: Деякі методи вимагають попереднього навчання алгоритму на підготовленому наборі даних, щоб створити модель кластеризації. Це може бути корисно, якщо ви маєте доступ до належної підготовки навчального набору даних, але може бути обтяжливим, якщо потрібно постійно навчати модель на нових даних.

Зазначені критерії допоможуть оцінити ефективність методів кластеризації текстових документів з різних позицій, включаючи точність, швидкість, роботу в реальному часі, гнучкість і зручність використання. При виборі конкретного методу кластеризації варто розглянути ці критерії і знайти баланс між ними залежно від потреб.

Існує безліч методів кластеризації документів, і багато з них можуть використовувати різні алгоритми для досягнення кластеризації. Нижче наведений список деяких методів, що використовуються для кластеризації документів:

1. K-means (k-середніх): Цей метод розподіляє документи на кластери шляхом знаходження k центроїдів, які представляють середні значення векторів документів у кожному кластері. Документи потім призначаються до найближчого центроїда згідно з їхніми векторними подібностями.

2. Hierarchical Clustering (ієрархічна кластеризація): Цей метод побудовує ієрархічну структуру кластерів шляхом об'єднання або розбиття кластерів на основі схожості документів. Він може бути агломеративним (об'єднує кожен документ в окремий кластер та об'єднує схожі кластери поступово) або дивізивним (починає з одного кластера, що містить всі документи, і поділяє його на менші кластери).

3. Latent Dirichlet Allocation (LDA): Цей статистичний метод моделює тематичну структуру колекції документів і використовує його для кластеризації. Він припускає, що кожен документ може містити декілька тем з певними ймовірностями, і кластеризує документи на основі спільних тематичних характеристик.

4. DBSCAN (Density-Based Spatial Clustering of Applications with Noise): Цей алгоритм кластеризації виявляє кластери на основі щільності точок у просторі. Він здатний виявляти кластери будь-якої форми та розміру, враховуючи наявність шуму та викидів.

5. Spectral Clustering: Цей метод використовує спектральну теорію графів для кластеризації. Він перетворює матрицю схожості документів на граф, виконує розклад Лапласа цього графу та застосовує кластеризацію до отриманих власних векторів.

6. Mean Shift: Цей метод шукає локальні максимуми щільності даних і рухається в напрямку більш щільних областей для знаходження центроїдів кластерів.

Ці алгоритми будуть детально проаналізовані і досліджені на відповідність критеріям рішення поставленої задачі кластеризації текстових документів.

2.1.1 K-means (k-середніх)

K - means - алгоритм в основі якого лежить ітеративний процес стабілізації центроїдів кластерів. Основною характеристикою кластера є його ценроїд і уся робота алгоритму спрямована на стабілізацію або, у кращому разі, повне припинення зміни центроїда кластера[8] [14].

Алгоритм складається з наступних кроків:

1. Вибираються початкові центроїди для безлічі документів. Існує декілька методик вибору центроїдів:
    1. Випадковий вибір: З безлічі документів випадковим чином вибирається k документів, де k - це задане число кластерів. Ці вибрані документи стають початковими центроїдами кластерів. Однак, у цьому методі необхідно явно вказувати необхідну кількість кластерів.
    2. Байєсівська оцінка: Використовується для визначення числа кластерів та їх центроїдів на основі Байєсовського підходу і апріорної інформації про розподіли в предметній області. Цей метод вимагає наявності апріорної інформації про природу даних, яка може бути використана для оцінки параметрів вибірки документів.
    3. Емпіричні оцінки: Використовуються для емпіричного визначення числа кластерів та їх центроїдів на основі аналізу вхідних даних. Ці методи дають найкращі результати в конкретній області, але не є універсальними і можуть варіюватись залежно від контексту застосування.
2. Всі документи з великої кількості розподіляються по кластерах. Кожен документ входить тільки в один кластер, той, у якого метрика близькості між центроїдом кластера і документом має найбільше значення.
3. після початкового розподілу документів по кластерах і визначення початкових центроїдів, перераховуються центроїди кластерів, виходячи з нового набору документів в кожному кластері. Цей процес виконується в циклі з наступними кроками:

1. Обчислення нових центроїдів: Для кожного кластера обчислюються нові центроїди на основі документів, які належать до цього кластера. Центроїд кластера може бути представлений як середнє значення векторів документів в цьому кластері або іншими методами обчислення центральної точки.
2. Перевірка стабільності центроїдів: Перевіряється, чи змінилися центроїди кластерів порівняно з попереднім кроком. Якщо центроїди перемістилися, то переходимо до пункту 2. В іншому випадку, якщо центроїди стабілізувалися або знаходяться в певному діапазоні, процес кластеризації завершується[14].

Оцінки обчислювальної складності алгоритму K-means залежать від кількості точок у наборі даних (n), кількості кластерів (k) і кількості ітерацій (t), необхідних для збіжності. Основні кроки алгоритму K-means включають ініціалізацію центроїдів, присвоєння точок до найближчих центроїдів і оновлення центроїдів. Оцінки складності для кожного кроку:

1. Ініціалізація центроїдів: Складність цієї операції залежить від використаної методики. У випадку звичайного випадкового вибору центроїдів складність становить O(k), де k - кількість кластерів.
2. Присвоєння точок до найближчих центроїдів: Кожній точці треба обчислити відстань до всіх центроїдів і вибрати найближчий. Ця операція має складність O(n * k), де n - кількість точок, а k - кількість кластерів.
3. Оновлення центроїдів: Нові центроїди обчислюються шляхом вирахування середнього значення точок у кожному кластері. Ця операція також має складність O(n * k), оскільки потрібно обробити всі точки для кожного кластеру.

Алгоритм K-means повторює кроки 2 і 3 до досягнення збіжності або до досягнення заданої кількості ітерацій (t). Таким чином, загальна складність алгоритму K-means оцінюється як O(t * n * k). Зазвичай кількість ітерацій є обмеженою, тому загальна складність може бути представлена як O(n * k * t).

Достоїнства методу:

- Лінійна швидкість роботи: Метод працює дуже швидко, що дозволяє досягти високої продуктивності в пошуковій системі.
- Використання значень матриці tf·idf: Використання цих значень дозволяє враховувати важливість термінів у документах при кластеризації, що поліпшує точність результатів.
- Метод не потребує навчання і може накопичувати зведення: Метод може працювати без необхідності навчання і, за необхідності, накопичувати знання для поліпшення точності кластеризації, зокрема за допомогою Байєсовських оцінок параметрів.

Недоліки методу:

- Потреба в заданні кількості кластерів: У початкових етапах методу потрібно явно вказати кількість кластерів, що може бути складно при відсутності апріорної інформації.
- Варіація результатів при випадковому виборі центроїдів: Якщо центроїди кластерів вибираються випадковим чином, результати можуть відрізнятися при повторних запусках на тій же вибірці документів. Це може бути викликано недостатньою якістю генератора випадкових чисел або рівномірним розподілом документів у просторі без явних зон згущення.
- Неінкрементний алгоритм: Метод не може ефективно обробляти нові документи або зміни в існуючих документах без повного повторного розрахунку кластерів.
- Неперетинання кластерів: Кластери, сформовані методом, не можуть перетинатися, що може бути обмеженням у випадках, коли документи мають багатогранні асоціації або належать до кількох тематичних груп одночасно.

Алгоритм k -средних (k - means clustering) - дуже швидкий, простій і досить точний метод кластеризації.

2.1.2 Hierarchical Clustering (ієрархічна кластеризація)

Ієрархічна кластеризація (Hierarchical Clustering) є одним із методів кластерного аналізу, який групує об'єкти у ієрархічну структуру класифікації. Вона використовує підхід "злиття" (agglomerative) або "розкладання" (divisive) для побудови цієї ієрархії.

У методі злиття, починаючи з кожного об'єкту як окремого кластера, послідовно об'єднуються найближчі кластери, поки всі об'єкти не будуть об'єднані в один великий кластер або досягнеся певний критерій зупинки. Цей процес зображується у вигляді дерева, відомого як дендрограма, де кожен вузол представляє кластер або об'єкт, а відстань між вузлами відображає ступінь віддаленості або схожості між ними[12]..

У методі розкладання, спочатку всі об'єкти належать до одного великого кластера, який потім рекурсивно розбивається на менші кластери досягнення певного критерію зупинки. Результатом є також дендрограма, але вона будується від найбільшого кластера до найменшого.

Один із ключових аспектів ієрархічної кластеризації - це можливість вибору рівня розбиття, що визначає кількість кластерів. Це дозволяє використовувати ієрархічну кластеризацію для різних потреб аналізу даних.

Алгоритм ієрархічної кластеризації може бути реалізований у двох варіантах: агломеративний (злиття) і дивізівний (розкладання). Розглянемо кожен з них:

1. Агломеративна ієрархічна кластеризація:
    1. На початку кожен об'єкт уявляється окремим кластером.
    2. Обчислюється матриця відстаней або схожості між кластерами (наприклад, за допомогою евклідової відстані або кореляції).
    3. Знаходяться два найближчих кластери за обраною метрикою.
    4. Ці два кластери об'єднуються в один новий кластер.
    5. Оновлюється матриця відстаней або схожості, враховуючи новий кластер.

6. Кроки 3-5 повторюються до отримання одного великого кластера, який містить всі об'єкти, або до досягнення заданого критерію зупинки.
7. Результат представляється у вигляді дендрограми, яка показує ієрархічну структуру кластерів.

2. Дивізівна ієрархічна кластеризація:
    1. Весь набір даних уявляється одним великим кластером.
    2. Кластер рекурсивно розбивається на менші кластери досягнення заданого критерію зупинки.
    3. Рекурсивний процес розбиття може використовувати різні методи, такі як розділення за медіаною, розділення за середнім значенням тощо.
    4. Кожен крок розбиття зображується у дендрограмі.

Обидва варіанти ієрархічної кластеризації мають свої переваги та недоліки. Вибір конкретного методу залежить від характеристик даних і вимог аналізу.

Оцінки обчислювальної складності для ієрархічної кластеризації (Hierarchical Clustering) залежать від використовуваного методу: агломеративного чи дивізивного.

1. Агломеративна ієрархічна кластеризація:
    - Обчислення відстаней: O(n^2) - обчислення відстаней між кожною парою точок, де n - кількість точок.
    - Побудова дерева злиття: O(n^3) - послідовне об'єднання кластерів до отримання повного дерева злиття.
2. Дивізивна ієрархічна кластеризація:
    - Обчислення відстаней: O(n^2) - обчислення відстаней між кожною парою точок, де n - кількість точок.
    - Рекурсивна декомпозиція: O(n^2 * log(n)) - рекурсивне ділення кластерів на підкластери з послідовним зменшенням кількості точок на кожному рівні.

Загальна оцінка складності для ієрархічної кластеризації залежить від кількості точок та використаного підходу (агломеративного або дивізивного). Також варто зазначити, що для великих наборів даних обчислювальна складність може бути значно вищою через велику кількість обчислень відстаней між точками.

Достоїнства ієрархічної кластеризації:

- Не вимагає попереднього визначення кількості кластерів.
- Забезпечує інтерпретовані результати у вигляді дендрограми.
- Дозволяє виявляти як великі, так і малі кластери.
- Зручний для візуалізації та аналізу ієрархічних зв'язків між кластерами.

Недоліки ієрархічної кластеризації:

- Вимагає значних обчислювальних ресурсів, особливо при великій кількості об'єктів.
- Не завжди ефективний для великих наборів даних.
- Деякі методи можуть бути чутливими до початкових умов або вибору метрики відстані.
- Потребує інтерпретації та визначення оптимального рівня розбиття на кластери.

2.1.3 Latent Dirichlet Allocation (LDA)

Прихований розподіл Діріхле (Latent Dirichlet Allocation, LDA) - це імовірнісна генеративна модель, що використовується для моделювання тем, яка є технікою для виявлення прихованих тем або тем у колекції документів. LDA припускає, що кожен документ є сумішшю різних тем, а кожна тема є розподілом по набору слів.

Основна мета LDA - виявити основну тематичну структуру в колекції документів, виводячи розподіл тем для кожного документа і розподіл слів для кожної теми. Модель передбачає наступний генеративний процес:

1. Для кожного документа:
   - Випадковий вибір розподілу за темами.

     Для кожного слова в документі:
     - Випадково вибирається тема з розподілу тем.
     - Випадковий вибір слова з розподілу слів теми.
2. Перебирайте документи і слова, щоб вивести приховану структуру тем.

Ключова ідея LDA полягає в тому, що спостережувані слова в документі можуть бути використані для виведення латентних (прихованих) тем, які породили ці слова. Аналізуючи частоту вживання слів у документах, LDA визначає найбільш ймовірні теми, які відображають зміст колекції документів[11].

Модель LDA вимагає вказівки кількості тем як параметра. Під час процесу виведення, який зазвичай використовує такі методи, як варіаційне виведення або вибірка Гіббса, модель оцінює розподіл тем для кожного документа і розподіл слів для кожної теми. Ці розподіли можна використовувати для призначення тем новим документам або для аналізу поширеності тем у колекції.

LDA має різні застосування, зокрема, для текстового аналізу, пошуку інформації, класифікації документів і рекомендаційних систем. Він дає змогу організовувати та узагальнювати великі колекції документів на основі їхнього тематичного змісту.

Оцінки обчислювальної складності для Latent Dirichlet Allocation (LDA) залежать від кількості документів (N), кількості слів у документі (M), кількості тем (K) та кількості ітерацій (T).

1. Побудова моделі:
    - Перевага EM-алгоритму в LDA полягає в тому, що він має лінійну залежність від кількості документів та слів, тобто його складність є O(N * M * K * T).
    - Найбільш обчислювально вимогливою є крок максимізації очікування (E-крок), де кожному слову документа призначається вага, що вимагає ітераційного обчислення ймовірностей усіх тем для кожного слова.
2. Перелік оцінок:
    - Побудова моделі: O(N * M * K * T)
    - Інференція тем: O(N * M * K * T * I), де I - кількість семплів для інференції, що використовується для покращення статистичної точності моделі.
    - Інференція нових документів: O(M * K * T), де M - кількість слів у новому документі.

Плюси LDA (Latent Dirichlet Allocation):

1. Неперевершена здатність до тематичного моделювання: LDA дозволяє виявити приховані теми або тематичні структури в колекції документів, що допомагає зрозуміти їх зміст і відношення між ними.
2. Гнучкість: LDA може бути використана для аналізу різних типів документів, включаючи тексти, звуки, зображення тощо. Вона також може бути застосована до різних завдань, таких як класифікація документів, пошук інформації та рекомендації.
3. Інтерпретованість результатів: LDA надає інтерпретовані результати, які можна пояснити і зрозуміти. Кожна тема представлена розподілом слів, що дозволяє встановити, які слова пов'язані з кожною темою.
4. Здатність до виявлення нових тем: LDA може виявляти нові теми в документах, які не були використані під час навчання моделі. Це робить LDA більш гнучким у порівнянні з попередньо заданими тематичними моделями.

Мінуси LDA:

1. Визначення кількості тем: LDA вимагає задання кількості тем в моделі. Вибір неправильної кількості тем може призвести до невірних або неадекватних результатів.
2. Чутливість до початкових значень: LDA може бути чутливим до початкових значень, з яких починається процес навчання моделі. Різні початкові значення можуть призвести до різних результатів.
3. Обчислювальна складність: LDA може бути обчислювально витратним, особливо для великих колекцій документів. Інференція може вимагати багато обчислень і займати значний час.
4. Втрата локальної структури: LDA не враховує локальну структуру документа, так як вважається, що порядок слів у документі не має значення. Це може призвести до втрати деякої інформації.

2.1.4 DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) - це алгоритм кластеризації, який використовує щільність точок у просторі для виявлення кластерів. Він є одним із найпопулярніших алгоритмів для кластеризації даних, особливо в задачах з неоднорідними розподілами густини[13].

Основні поняття, пов'язані з DBSCAN:

1. Щільність (Density): Щільність точки визначається як кількість інших точок, що знаходяться у встановленому радіусі навколо неї. Це відображає, наскільки точка щільно оточена іншими точками.
2. Epsilon (ε): Це параметр алгоритму, що визначає радіус навколо кожної точки, в якому шукаються інші точки для визначення її щільності.
3. MinPts: Це ще один параметр алгоритму, який визначає мінімальну кількість точок, необхідних у радіусі ε для того, щоб точка була визнана як ядро (core point).

Процес роботи алгоритму DBSCAN:

1. Вибір початкової точки, яку ще не було відвідано.
2. Визначення щільності цієї точки, тобто підрахунок кількості точок, які знаходяться у встановленому радіусі ε навколо неї.
3. Якщо щільність точки перевищує або дорівнює MinPts, то ця точка визначається як ядро (core point) і включається до кластера. Всі точки, які знаходяться у радіусі ε навколо цієї точки і мають достатню щільність, також включаються до цього кластера.
4. Продовження процесу для кожної нової точки, що була додана до кластера, включаючи виявлення і додавання її сусідів, якщо вони відповідають умові щільності.
5. Повторення кроків 1-4 для інших невідвіданих точок, поки не будуть відвідані всі точки.
6. Точки, які не мають достатньої щільності для входження в будь-який кластер, вважаються шумом або викидаються.

Оцінки обчислювальної складності для DBSCAN (Density-Based Spatial Clustering of Applications with Noise) залежать від кількості точок у вхідному наборі даних (N) та параметрів алгоритму, таких як радіус епсилон (ε) і мінімальна кількість сусідів (MinPts).

1. Побудова кластерів:
    - У загальному випадку, для кожної точки вхідного набору даних потрібно обчислити відстань до всіх інших точок, що має складність O(N^2).

- При використанні підходу зі структурами даних, такими як k-d дерева або R-дерева, можна досягти складності O(N log N) для пошуку сусідів.

2. Кластеризація:
    - DBSCAN використовує пошук сусідів і формування кластерів, що залежить від кількості точок і їх взаємних відносин. Оцінка складності зазвичай є O(N log N) або O(N^2), залежно від використовуваних структур даних та методів пошуку сусідів.

Достоїнства DBSCAN:

- Незалежний від форми кластерів: DBSCAN може ефективно виявляти кластери будь-якої форми, так як він базується на щільності точок, а не на геометричних формах.
- Здатність виявляти шум: DBSCAN може ідентифікувати та видалити шумові точки, які не входять до жодного кластера.
- Масштабованість: DBSCAN може ефективно працювати з великими наборами даних.

Недоліки DBSCAN:

- Чутливість до параметрів: DBSCAN має параметри ε і MinPts, які потрібно налаштовувати. Вибір неправильних значень цих параметрів може призвести до некоректних результатів.
- Потребує визначення кількості кластерів: DBSCAN не визначає автоматично кількість кластерів, тому ця інформація повинна бути визначена користувачем або шукатися за допомогою інших методів.
- Чутливість до щільності: При неоднорідних щільностях у просторі можуть виникати проблеми з визначенням правильних кластерів.

DBSCAN - корисний алгоритм кластеризації, який використовує щільність точок у просторі для виявлення кластерів без залежності від їх форми та розміру.

2.1.5 Mean Shift

Mean Shift є алгоритмом кластеризації, який шукає локальні екстремуми функції щільності даних для визначення кластерів. Основна ідея алгоритму полягає в зсуві кожної точки в напрямку збільшення густини даних до досягнення локального максимуму густини.

Основні кроки алгоритму Mean Shift:

1. Вибір початкових центроїдів: Початкові центроїди можуть бути вибрані випадковим чином або на основі попередніх знань про дані.

2. Обчислення вагових середніх значень: Для кожного зразка обчислюється вагове середнє значення, використовуючи ядро або функцію подібності. Це вагове середнє значення представляється як нове положення точки.

3. Зсув точок: Кожна точка зсувається до її вагового середнього значення.

4. Повторення процесу: Кроки 2 і 3 повторюються до досягнення збіжності, тобто поки точки більше не зсуваються або зсув менше певного порогового значення.

Алгоритм продовжує виконуватися, поки всі точки не стабілізуються в околиці локальних максимумів густини, що представляють кластери. Кластери визначаються таким чином, що точки, які стабілізуються недалеко одна від одної, вважаються належними до одного кластера.

Оцінки обчислювальної складності для алгоритму Mean Shift залежать від кількості точок у вхідному наборі даних (N) та параметрів алгоритму, таких як радіус смішення (bandwidth).

1. Побудова ядерного графа:
    - У загальному випадку, для кожної точки вхідного набору даних потрібно обчислити відстань до всіх інших точок, що має складність O(N^2).
    - При використанні підходів зі структурами даних, такими як k-d дерева або R-дерева, можна досягти складності O(N log N) для пошуку сусідів.
2. Знаходження центроїдів:
    - Для кожної точки вхідного набору даних потрібно виконати ітеративний процес зсуву в напрямку градієнта, що залежить від кількості точок і їх взаємних відносин.
    - Оцінка складності зазвичай є O(N^2) або O(N^3), в залежності від методів обчислення градієнта та використаної оптимізації.

Переваги алгоритму Mean Shift:

- Не вимагає заздалегідного визначення кількості кластерів.
- Добре працює з даними складної форми та розмірів кластерів.
- Виявляє кластери різної густини та їхні центри.

Недоліки алгоритму Mean Shift:

- Вимагає налагодження параметрів, таких як радіус ядра або функція подібності.
- Може бути обчислювально вимогливим, особливо при роботі з великими наборами даних.
- Чутливий до початкового вибору центроїдів.

У загальному, Mean Shift є потужним алгоритмом кластеризації, здатним працювати з різноманітними типами даних і виявляти кластери без вимог до кількості кластерів. Однак, для досягнення оптимальних результатів важливо налагодити його параметри і врахувати обчислювальні обмеження.

2.2 Вибір методів оптимізації пошуку документів в мережі Інтернет з використанням методів кластеризації

Ми розглянули обрані методи кластерного аналізу тепер тезисно порівняємо їх:

1. K-means (k-середніх):
    - Використовується для розділення об'єктів на кластери шляхом знаходження центру кожного кластера.
    - Вимагає попередньо визначеної кількості кластерів.
    - Добре підходить для розділення документів за їхньою темою або спільними ключовими словами.
    - Швидко працює на великих наборах даних, але може залежати від початкових точок центрів кластерів.
2. Hierarchical Clustering (ієрархічна кластеризація):
    - Створює ієрархічну структуру кластерів, яка може бути представлена у вигляді дерева (дендрограми).
    - Не потребує попереднього визначення кількості кластерів.
    - Дозволяє візуалізувати взаємозв'язки між кластерами та документами.
    - Може бути витратним з точки зору обчислювальних ресурсів при роботі з великими наборами даних.
3. Latent Dirichlet Allocation (LDA):
    - Використовується для тематичної моделювання тексту шляхом виявлення "латентних" тем, які відображаються у вхідних документах.
    - Дозволяє кластеризувати документи за їхніми тематиками та визначити імовірність належності до кожної теми.
    - Зручний для пошуку документів за темою або контекстом.
    - Вимагає попереднього моделювання тем та оптимізації параметрів.
4. DBSCAN (Density-Based Spatial Clustering of Applications with Noise):
    - Базується на густині точок у просторі для розділення документів на кластери.
    - Може виявити кластери будь-якої форми та виявити шумові точки.
    - Не потребує попереднього визначення кількості кластерів.
    - Працює добре для виявлення груп документів з високою густиною.
5. Mean Shift:
    - Шукає локальні максимуми густини документів, щоб визначити центри кластерів.
    - Не потребує попереднього визначення кількості кластерів.

- Добре підходить для кластеризації документів з різними розмірами та формами.
- Може бути вимогливим до обчислювальних ресурсів при роботі з великими наборами даних.

Кожен з цих методів має свої переваги та обмеження, типи даних та обсяг інформації. При реалізації пошуку документів в інформаційно-пошукових системах, може бути цікавим комбінувати декілька методів для досягнення кращих результатів, але у данній роботі ми розглянемо їх окремо.

У таблиці 2.2 наведено характеристики обраних для аналізу методів кластерного аналізу.

Ми порівняємо методи K-means і LDA у контексті пошуку документів в інформаційно-пошукових системах, оскільки обидва методи зосереджені на кластеризації документів за їхньою схожістю або тематикою.

K-means використовується для розділення документів на попередньо визначену кількість кластерів, знаходячи центри кластерів. Цей метод підходить для групування документів за спільними ключовими словами або темою. Він простий у реалізації та швидкий на великих обсягах даних. Однак, його обмеженням є необхідність попереднього визначення кількості кластерів.

LDA (Latent Dirichlet Allocation) використовується для тематичного моделювання документів. Він дозволяє виявити "латентні" теми, які відображаються в документах. LDA може бути використаний для кластеризації документів за їхніми тематиками і визначенням ймовірності належності до кожної теми. Цей метод дозволяє здійснити гнучку кластеризацію без необхідності визначення кількості кластерів. Однак, його використання вимагає попереднього моделювання тем та оптимізації параметрів.

Порівняємо метод Latent Dirichlet Allocation (LDA) і метод K-means за кількома аспектами:

1. Вид методу:
    - LDA: Латентний метод, що використовуємо для тематичної моделювання текстів.
    - K-means: Числовий метод кластеризації, що базується на відстані між об'єктами.
2. Обмеження:
    - LDA: Немає обмежень стосовно кількості кластерів, але потрібно визначити кількість тем.
    - K-means: Вимагає задання кількості кластерів та початкових центроїдів.
3. Перетин кластерів:

- LDA: Ієрархічність кластерів відсутня, кожен документ може належати до кількох тем.
- K-means: Кластери є взаємно виключними, кожен об'єкт належить лише до одного кластера.

4. Інкрементність алгоритму:
   - LDA: Залежить від підходу, але може бути важко додавати нові дані або оновлювати модель з надходженням нових документів.
   - K-means: Зазвичай не є інкрементним, потрібно повністю перераховувати кластери при зміні даних.
5. Використовані числові характеристики документів:
   - LDA: Використовує "Bag-of-words" представлення документів, яке враховує частоту термінів у текстах.
   - K-means: Використовує числові ознаки документів, наприклад, tf-idf, що описують важливість слів у документах.
6. Попереднє навчання:
   - LDA: Потрібно провести навчання моделі, яке може вимагати більше обчислювальних ресурсів та часу.
   - K-means: Навчання проводиться безпосередньо на даних без потреби в додатковому попередньому навчанні.
7. Швидкість роботи:
   - LDA: Швидкість залежить від розміру документів та кількості тем, може бути повільнішою для великих корпусів текстів.
   - K-means: Швидкість роботи залежить від кількості документів, кількості кластерів та кількості ітерацій.

Порівняння цих двох методів дозволяє оцінити їхні переваги та недоліки в контексті пошуку документів в інформаційно-пошукових системах. Обидва методи можуть бути корисними для групування документів, але з різними підходами та вимогами.

Таблиця 2.2 - Таблиця порівняння технічних характеристик алгоритмів

Метод	Вид методу	Обмеження	Перетин кластерів	Інкрементність алгоритму	Використовувані числові характеристики документів	Попереднє навчання	Швидкість роботи
K-means (k-середніх)	Числовий	Вимагає задання кількості кластерів та початкових центроїдів	Немає вбудованого механізму для перетину кластерів	-	tf-idf	-	O(knT), де n – кількість документів, k – кількість кластерів, T – кількість ітерацій
Hierarchical Clustering (ієрархічна кластеризація)	Нечисловий	Немає обмежень	+	+	Схожість між об'єктами	-	O(n^2 log n), де n – кількість об'єктів
Latent Dirichlet Allocation (LDA)	Числовий	Немає обмежень	-	+	Bag-of-words представлення документів	+	Залежить від розміру документів та кількості тем
DBSCAN (Density-Based Spatial Clustering of Applications with Noise)	Нечисловий	Не вимагає задання кількості кластерів	-	+	Необхідно обчислити схожість між об'єктами	-	O(n log n), де n – кількість об'єктів
Mean Shift	Числовий	Не вимагає задання кількості кластерів	-	+	Необхідно обчислити схожість між об'єктами	-	Залежить від розміру даних та параметрів алгоритму

2.3 Кластеризація множин з використанням алгоритму K-means

Нами було проведено дослідження декількох популярних методів кластеризації, і алгоритм K-means був визнаний одним з найбільш відповідних для розбиття колекції текстових документів на групи. Алгоритм K-means базується на ітеративному процесі стабілізації центроїдів кластерів. Основна характеристика кластера - його центроїд, і алгоритм спрямований на стабілізацію центроїда шляхом мінімізації середньоквадратичного відхилення точок в кожному кластері.

Алгоритм K-means будує k кластерів, які можуть бути розташовані на значній відстані один від одного. Дія алгоритму полягає у тому, щоб мінімізувати середньоквадратичне відхилення точок у кожному кластері:

$$V = \sum_{i=1}^{k} \sum_{x_j \in S_i} (x_j - \mu_i)^2,$$

де k - кількість кластерів, S_i – отримані кластери, $i = 1,2 \dots, k$ и μ_i - центри мас векторів $x_j \in S_i$.

Процес включає в себе початкову вибірку центроїдів, призначення точок до найближчого кластера, перерахування центроїдів і повторення цих кроків до досягнення стабільної конфігурації кластерів[14].

В контексті пошуку документів в інтернеті алгоритм K-means може бути використаний для групування схожих документів за їхнім змістом або іншими властивостями. Наприклад, документи, що містять подібні ключові слова або терміни, можуть бути об'єднані в один кластер. Це дозволяє зменшити обсяг пошуку, спрямованого на конкретні кластери, замість виконання пошуку в усіх документах.

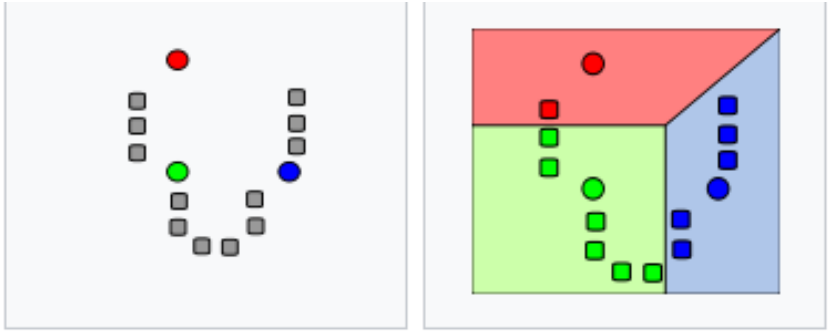
Основна ідея алгоритму K-means полягає в перерахуванні центрів мас для кожного кластера на кожній ітерації. Вектори даних розподіляються у кластери знову, враховуючи найближчі центри за обраною метрикою. Алгоритм завершується, коли на якійсь ітерації не відбувається зміни кластерів. Основне завдання, що вирішується алгоритмом K-means, - це визначення кластерів з попередньо встановленим числом, які максимально відрізняються один від одного за середніми значеннями.

Загальна ідея алгоритму полягає у тому, що задане фіксоване число k кластерів спостереження призначаються кластерам таким чином, щоб середні значення у кожному кластері (для всіх змінних) максимально відрізнялися одне від одного. Обмеження алгоритму K-means полягає в тому, що він показує найкращі результати при використанні невеликих обсягів даних.

Таким чином, основні ідеї алгоритму K-means, які були описані, підтримують використання цього методу для розбиття колекції текстових

документів на групи в процесі пошуку документів в інтернеті. Алгоритм K-means дозволяє групувати документи за їхнім змістом, що полегшує пошук і відбір необхідних документів для користувачів.

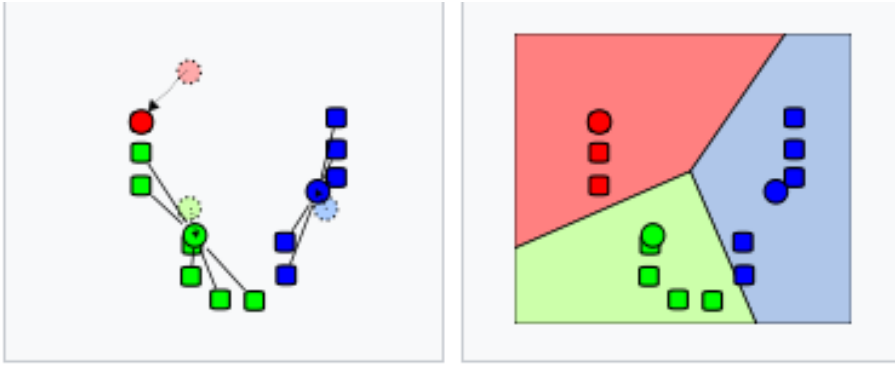
Демонстрація дії алгоритму представлена на рисунку 2.3.1 і 2.3.2 . Початкові точки вибрані випадково. Рисунок взято із статті[15].



1. *k* початкових «середніх» (тут *k*=3) випадково згенеровані у межах домени даних (кольорові).

2. створено *k* кластерів, асоціюючи кожне спостереження з найближчим середнім. Розбиття відбувається згідно з діаграмою Вороного утвореною середніми.

Рисунок 2.3.1 - Демонстрація дії алгоритму Kmeans



3. Центроїд кожного з *k* кластерів стає новим середнім.

4. Кроки 2 і 3 повторюються до досягнення збіжності.

Рисунок 2.3.2 - Демонстрація дії алгоритму Kmeans (продовження)

2.4 Побудова математичної моделі кластерізації з використанням алгоритму K-means

Перед побудовою математичної моделі буде наведено детальний опис алгоритму.

На першому етапі алгоритму K-means проводиться початковий розподіл об'єктів по кластерах. Це досягається шляхом вибору числа k - кількості кластерів. На цьому етапі об'єкти, представлені векторами у вхідній матриці X, використовуються як початкові центри кластерів. Кожен об'єкт призначається до кластера згідно з його найближчим центроїдом.

Існують різні методи вибору початкових центроїдів:

1. Випадковий вибір: У даному методі вибору початкових центроїдів для кластерів випадковим чином вибирається k документів з наявного набору документів. Кількість вибраних документів k повинна відповідати необхідній кількості кластерів. На першому кроці ці вибрані документи стають початковими центроїдами кластерів. Цей метод передбачає, що користувач вже заздалегідь знає, скільки кластерів необхідно сформувати.
2. Байєсовська оцінка: Вибір початкових центроїдів на основі Байєсовської оцінки[14] передбачає використання статистичного підходу для оцінювання та визначення відповідної кількості кластерів і їх центроїдів на основі безлічі документів. Цей підхід ґрунтується на Байєсовській статистиці, яка використовує апріорну інформацію про природу та розподіли даних, що переважно мають місце в предметній області [14].Якщо в наявності немає апріорної інформації, то можна вважати, що розподіл даних є нормальним (Гаусовим) або рівномірним. За допомогою Байєсовської оцінки та апріорної інформації можна проводити оцінку параметрів даних і визначати оптимальну кількість кластерів і їх центроїдів. Використовуючи статистичні методи, можна знаходити базові статистики, оцінювати розподіли даних і вибирати найбільш ймовірну кількість кластерів.
3. Емпіричні оцінки: застосовуються емпіричні методи оцінки кількості кластерів та їх центроїдів, які базуються на досвіді та знаннях про конкретну область даних. Ці методи можуть давати кращі результати в конкретних ситуаціях, але не завжди є універсальними.

На другому етапі, відомому як ітеративний перерозподіл, об'єкти (у даному випадку документи) розподіляються по кластерах шляхом обчислення відстані між об'єктом і центроїдами кластерів і вибору найменшої відстані. Усі документи великої кількості розподіляються серед кластерів, і кожен документ

потрапляє тільки в один кластер, а саме в той кластер, метрика близькості центроїда якого і документу має найбільше значення.

На третьому етапі після розподілу всіх об'єктів по кластерах, проводиться обчислення нових центрів кластерів. Центри кластерів вважаються покоординатними середніми значеннями ознак всіх об'єктів, що належать до кожного кластера.

$$\mu_i = \frac{\sum_{k=1}^{K_i} x_k}{K_i},$$

K_i – кількість векторів у і-ому кластері, x_k – вектор матриці X.

На четвертому етапі при $\mu_i = \mu_{i-1}$, то кластерні центри стабілізувалися і відповідно розподіл закінчений. Інакше переходиться до першого етапу.

Математична модель грунтується на загальній ідеї алгоритму : мінімізація відстаней між об'єктами в кластерах. Зупинка відбувається, коли мінімізувати відстані більше вже неможливо.

Цільова функція: $V = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 \to \min$, де $\left\| x_i^{(j)} - c_j \right\|^2$ - вибрана міра відстаней між точкою $x_i^{(j)}$ та ценром кластеру c_j, і є індикатором відстані n точок від центрів їх кластерів, k - число кластерів; x_i – вектор (рядок) матриці X, а як матриця X прийнята матриця $tf * idf$:

$$tf * idf = \frac{n_i}{\sum_k n_k} * \log \frac{|D|}{|(d_i \ni t_i)|},$$

де n_i - є число входжень слова в документ, а в знаменнику - загальне число слів в цьому документі, |D| - кількість документів в корпусі, $|(d_i \ni t_i)|$- кількість документів, в яких зустрічається t_i (коли $n_i \neq 0$).

Вибір оптимального числа кластерів є складним завданням і може залежати від природи даних і мети кластеризації. Якщо немає конкретних припущень щодо кількості кластерів, рекомендується виконати кластеризацію для різних значень кількості кластерів і порівняти отримані результати.

Один з підходів до перевірки якості кластеризації - це обчислення середніх значень для кожного кластера і оцінка, наскільки ці середні значення відрізняються між кластерами. Якщо кластери є хорошо відрізняються один від одного, то очікується, що середні значення для різних кластерів будуть суттєво різними для більшості або всіх ознак.

2.5 Кластеризація множин з використанням алгоритму Latent Dirichlet Allocation

В результаті дослідження в попередній частині роботи алгоритм LDA (Latent Dirichlet Allocation) був вибраний як один з найбільш оптимальних для розбиття колекції текстових документів на тематичні групи. Розглянемо цей алгоритм детальніше [16].

LDA є ймовірнісним генеративним моделюванням, яке дозволяє виявити латентні (приховані) теми в колекції документів. Алгоритм LDA базується на припущенні, що кожен документ можна розглядати як мішок слів, а коже слово може належати до різних тем.

Латентний Дирихле-алокаційний процес (LDA) є статистичною моделлю, яка використовується для аналізу тематичної структури колекції текстових документів. В LDA припускається, що кожен документ є сукупністю деякої кількості тем, а кожна тема є розподілом слів. Основною метою LDA є виявлення цих тем і приналежності слів до них.

Ідея LDA належить Девіду Блею та Андрю Нгуєну (2003 рік). У моделі LDA припускається, що кожен документ може містити декілька тем з певними ймовірностями. При цьому сліва в документах розподіляються за певними тематичними розподілами. Модель LDA базується на Байєсовській ймовірнісній інтерпретації тематичної структури.

Використовуючи модель LDA, можна отримати розподіли тем у документах та розподіли слів у темах. Це дає можливість здійснювати тематичний аналіз колекцій документів, виявляти семантичні зв'язки і здійснювати категоризацію текстових даних. LDA знаходить широке застосування у сфері обробки природних мов, інформаційного пошуку, рекомендаційних системах та інших задачах аналізу тексту.

LDA (Latent Dirichlet Allocation) є потужним інструментом для вирішення різних завдань аналізу тексту і розуміння його тематичної структури. Основне завдання, яке може бути вирішене за допомогою LDA, включає:

1. Тематичний аналіз: LDA дозволяє виявляти теми, які присутні в колекції текстових документів. Він дозволяє встановити ймовірнісний розподіл тем для кожного документа і розподіл слів для кожної теми. Це допомагає зрозуміти основні теми, які присутні в текстових даних.
2. Кластеризація тексту: Використовуючи LDA, можна кластеризувати текстові документи за їхньою тематикою. Документи, які мають схожі розподіли тем, можуть бути груповані разом, що дозволяє створювати тематичні кластери.
3. Рекомендаційні системи: LDA може використовуватись для рекомендаційних систем, де базуючись на тематичному аналізі

документів, можна рекомендувати користувачам схожі документи або продукти, які відповідають їхнім інтересам.

4. Аналіз сентименту: LDA може бути використаний для аналізу сентименту в текстових документах. Шляхом виявлення тем та їх асоціації з позитивним або негативним змістом, можна оцінювати сентимент тексту.
5. Розрізнення авторства: LDA може допомогти в розрізненні авторства текстових документів, виявляючи унікальні розподіли тем, пов'язані з кожним автором.
6. Пошук інформації: LDA може використовуватися для покращення пошукових систем шляхом асоціювання тематичних слів з документами та виявлен анням семантичних зв'язків між документами.

LDA є гнучким інструментом, який може бути застосований до різноманітних завдань аналізу тексту залежно від потреб дослідника або додатку. Він дозволяє отримувати важливу інформацію про структуру документів і тематичні залежності, що сприяє покращенню розуміння текстових даних і вирішенню конкретних завдань аналізу.

Модель LDA базується на припущенні, що кожен документ можна представити як комбінацію різних тем, а кожа тема може бути представлена як розподілм слова.

Структура LDA включає наступні компоненти:

1. Документи: Аналізувані текстові документи складають початкову колекцію даних. Кожен документ може містити довільну кількість слів.
2. Теми: Теми є складовою частиною моделі LDA і представляють тематичні концепції або ідеї, які присутні в текстових документах. Наприклад, для колекції новинних статей можуть бути такі теми, як "політика", "спорт", "фінанси" тощо. Кількість тем може бути задана перед запуском моделі LDA.
3. Слова: Кожен документ складається зі слова. Слова визначаються в контексті теми, до якої вони відносяться. Модель LDA вважає, що коже слово в документі вибирається з певного розподілу ймовірностей, пов'язаного з темою документа та розподілом слів для цієї теми.
4. Гіперпараметри: LDA має декілька гіперпараметрів, які потрібно задати перед навчанням моделі. Ці параметри включають кількість тем, розподіл Діріхле для тем та розподіл Діріхле для слів.
5. Інференція: Навчання моделі LDA включає процес інференції, в якому з використанням байєсівського підходу визначаються розподіли тем для кожого документа та розподіли слів для кожої теми. Цей процес включає

ітеративне оновлення розподілів на підставі спостережень у вхідних даних.

6. Вихідні результати: Після завершення навчання моделі LDA, можна отримати розподіли тем для кожного документа та розподіли слів для кожної теми. Ці результати можуть бути використані для подальшого аналізу, кластеризації документів, рекомендацій тощо.

Структура LDA враховує взаємозв'язок між документами, темами і словами, що дозволяє виявляти тематичну структуру інформації в текстових документах.

Модель LDA (Latent Dirichlet Allocation) навчається шляхом ітеративного процесу, який включає наступні кроки:

1. Початкова ініціалізація: Встановлюються початкові значення параметрів моделі, таких як кількість тем, розподіли тем у документах та розподіли слів у темах.
2. Навчання емпіричного розподілу: На першому кроці навчання модель використовує емпіричний розподіл, де кожне слово у документі призначається випадковій темі. Це створює початкове припущення про теми в документах.
3. Оновлення розподілів: В цьому кроці модель оновлює розподіли тем у документах та розподіли слів у темах на основі припущення, зробленого на попередньому кроці. Використовуючи статистичні методи, модель враховує частоту виникнення слів у темах та документах для покращення розподілів.
4. Повторення кроків 2-3: Кроки 2 і 3 повторюються декілька разів або до тих пір, поки модель не збіжиться до стабільних значень розподілів тем і слів.

Після завершення навчання модель LDA може бути використана для інференції розподілу тем у нових документах або для виконання інших завдань аналізу тексту, таких як кластеризація, аналіз сентименту тощо.

Важливо відзначити, що навчання моделі LDA є обчислювально складним процесом, особливо при великій кількості документів та слів.

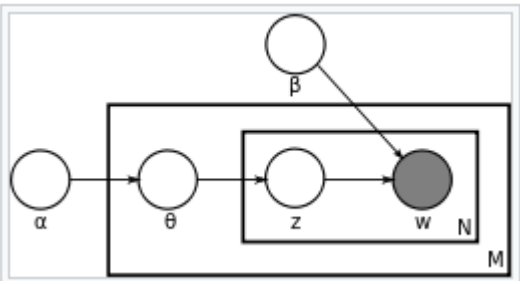


Рисунок 2.5.1 Нотація пластини[17], що представляє модель LDA

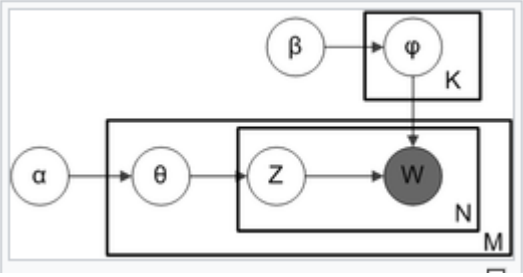


Рисунок 2.5.2 Нотація пластин[17], для LDA з розподіленими Діріхле розподіленими розподілами тем-слів

M позначає кількість документів

N - кількість слів у даному документі (документ *i* має N_i слова)

α — параметр пріора Діріхле для розподілу тем для кожного документа

β — параметр пріора Діріхле для розподілу слів за темою

θ_i Розподіл теми для *документа I*

φ_k — розподіл слів для теми *K*

z_{ij} є темою для *j-го* слова в документі *I*

w_{ij} є конкретним словом.

Рисунок 2.5.3 Позначення на схемах вище[18],

2.6 Побудова математичної моделі кластерізації з використанням алгоритму Latent Dirichlet Allocation

Розглянемо роботу алгоритму кластеризації з використанням Latent Dirichlet Allocation[19].

Спрощений алгоритм роботи LDA (Latent Dirichlet Allocation) може бути наступним:

1. Підготовка даних:
    - Зберіть колекцію текстових документів, з якою ви будете працювати.
    - Виконайте попередню обробку даних, таку як токенізація (розбиття тексту на окремі слова або токени), видалення стоп-слів та лематизація (перетворення слів до базової форми).
2. Визначення параметрів моделі:
    - Визначте кількість тем, яку ви хочете виявити в текстових документах.
    - Встановіть гіперпараметри моделі, такі як альфа і бета. Альфа відповідає за розподіл тем у документі, а бета - за розподіл слів у темі.
3. Ініціалізація:
    - Ініціалізуйте матриці тем та слово-темностей для кожного документа в колекції.
    - Випадково призначте кожному слову в документі початкову тему.
4. Ітерації навчання:
    - Для кожного документа:
        - Для кожного слова в документі:
            - Обновіть локальні лічильники темностей для слова та теми.
            - Застосуйте формули Байєса для оновлення темностей слів та тем.
    - Після побудови локальних лічильників в усіх документах, виконайте глобальне оновлення матриць тем та слово-темностей.
5. Повторюйте крок 4 протягом заданої кількості ітерацій або до досягнення збіжності моделі.
6. Оцінка моделі:
    - Виміряйте перплексію (perplexity) моделі, щоб оцінити її якість. Менша значення перплексії вказує на кращу модель.

Розглянемо принципи роботи LDA на прикладі кластеризації 5ти документів:

Прихований розподіл Діріхле (LDA) є інструментом і методом для моделювання тем в текстових документах. Використовуючи LDA, документи класифікуються або категоризуються за темами, а слова моделюються на основі розподілів тем та розподілів слів.

Основна ідея LDA полягає у двох ключових припущеннях:

1. Документи є сумішшю тем: Кожен документ вважається сумішшю різних тем. Наприклад, документ може містити як спортивні, так і політичні теми, але з різною інтенсивністю.
2. Теми є сумішшю токенів (слів): Кожна тема вважається сумішшю різних слів з певними ймовірностями. Наприклад, тема "спорт" може містити слова, пов'язані з футболом, баскетболом, тенісом тощо, але з різними ймовірностями появи.

І ці теми, використовуючи розподіл ймовірностей, генерують слова. Статистичною мовою документи називаються щільністю ймовірності (або розподілом) тем, а теми - щільністю ймовірності (або розподілом) слів.

LDA застосовує два важливих припущення до заданого корпусу. Припустимо, у нас є корпус з наступними п'ятьма документами:

Документ 1: Я хочу подивитися фільм на цих вихідних.

Документ 2: Я вчора ходив по магазинах. Нова Зеландія виграла чемпіонат світу з тесту, перемігши Індію у Саутгемптоні з різницею у вісім м'ячів.

Документ 3: Я не дивлюся крикет. На Netflix і Amazon Prime є дуже хороші фільми для перегляду.

Документ 4: Фільми - це гарний спосіб розслабитися, але цього разу я хотів би помалювати і почитати кілька хороших книжок. Це було так давно!

Документ 5: Цей чорничний молочний коктейль такий смачний! Спробуйте почитати книги доктора Джо Діспенза. Його роботи так змінюють правила гри! Його книги допомогли дізнатися так багато про те, як наші думки впливають на нашу біологію і як ми всі можемо перепрограмувати наш мозок.

Будь-який корпус, тобто колекцію документів, можна представити у вигляді матриці "документ-слово" (або "документ-термін"), також відомої як DTM.

Ми знаємо, що першим кроком з текстовими даними є очищення, попередня обробка та токенізація тексту до слів. Після попередньої обробки документів ми отримуємо наступну матрицю слів, де

D1, D2, D3, D4 і D5 - це п'ять документів, а слова позначені буквами W, скажімо, є 8 унікальних слів від W1 до W8.

Таким чином, матриця має форму 5 * 8 (п'ять рядків і вісім стовпців)(рисунок 2.6.1):

	W1	W2	W3	W4	W5	W6	W7	W8
D1	0	1	1	0	1	1	0	1
D2	1	1	1	1	0	1	1	0
D3	1	0	0	0	1	0	0	1
D4	1	1	0	1	0	0	1	0
D5	0	1	0	1	0	0	1	0

Рисунок 2.6.1: Матриця слів у документах

Отже, зараз корпус - це переважно попередньо оброблена матриця документ-слово, в якій кожен рядок - це документ, а кожен стовпчик - це лексеми або слова.

LDA перетворює цю матрицю документ-слово на дві інші матриці: Матрицю термінів документа і Матрицю слів теми, як показанона рисунку2.6.2:

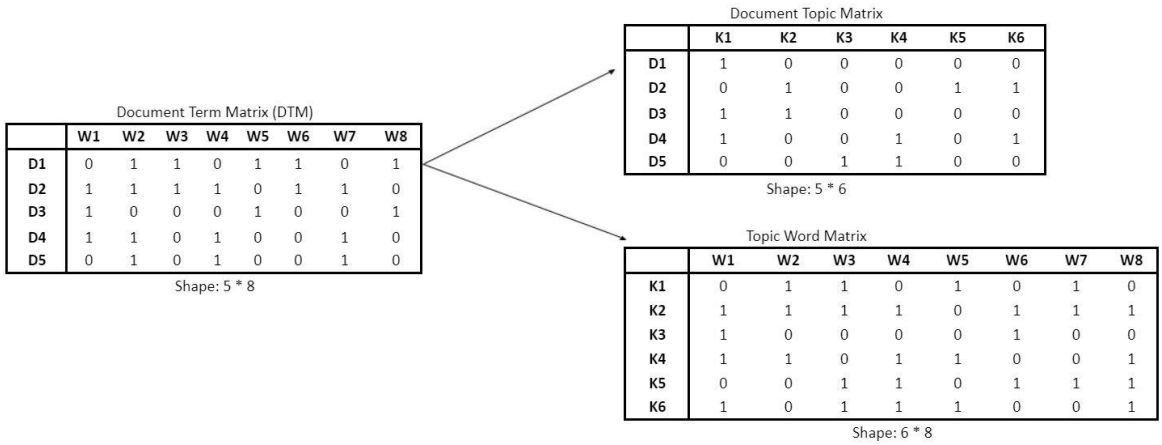


Рисунок 2.6.2: Матриця термінів документа і Матриця слів теми

Ці матриці:

Матриця Документ-Тема має розмірність 5x6, оскільки ми маємо 5 документів і 6 тем. Кожен елемент цієї матриці показує ймовірність належності документа до певної теми.

З іншого боку, матриця Тема-Слово має розмірність 5x8, оскільки у нас є 5 тем і 8 унікальних лексем (слів) у словнику. Кожен елемент цієї матриці показує ймовірність використання певного слова в певній темі.

В результаті навчання моделі LDA, ми отримуємо ці дві матриці, які відображають відношення між документами, темами і словами. Вони дозволяють нам розуміти, як теми розподіляються у документах та які слова характеризують кожну тему.

Розглянемо векторний простір LDA

Весь простір LDA та його набір даних представлено на рисунку 2.6.3:

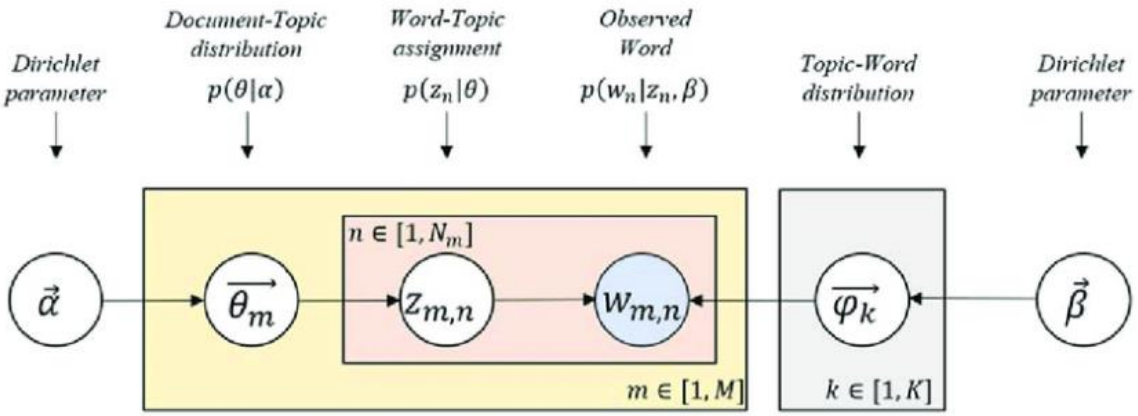


Рисунок 2.6.3: Простір LDA та його набір даних[21]

- M: загальна кількість документів у корпусі
- N: кількість слів у документі
- w: слово в документі
- z: латентна тема, призначена слову
- тета (θ): розподіл тем
- Параметри LDA моделі s: Альфа (α) та Бета (β)

Жовта рамка позначає всі документи в корпусі (позначені M). У нашому випадку M = 5, оскільки ми маємо 5 документів.

Далі, персиковий блок - це кількість слів у документі, задана через N

Усередині цієї персикової блоку може бути багато слів. Одним з таких слів є слово w, яке знаходиться у синьому кружечку.

Згідно з LDA, кожне слово асоціюється (або пов'язане) з латентною (або прихованою) темою, яка тут позначена Z. Тепер, це присвоєння Z слову-темі в цих документах дає розподіл слів-тем, присутніх у корпусі, який представлений тетою (θ).

Модель LDA має два параметри, які керують розподілом:

- Альфа (α) контролює розподіл тем у документі,
- Бета (β) керує розподілом слів у темі.

Кінцева мета LDA - знайти найоптимальніше представлення матриці Документ-Тема і матриці Тема-Слово, щоб знайти найоптимальніший розподіл Документ-Тема і Тема-Слово.

Оскільки LDA припускає, що документи є сумішшю тем, а теми є сумішшю слів, то LDA повертається з рівня документів, щоб визначити, які теми могли б породити ці документи і які слова могли б породити ці теми.

Тепер візьмемо наш корпус, який складається з 5 документів (D1 - D5) і відповідної кількості слів:

D1 = (w1, w2, w3, w4, w5, w6, w7, w8)

D2 = (w`1, w`2, w`3, w`4, w`5, w`6, w`7, w`8, w`9, w`10)

D3 = (w`1, w`2, w`3, w`4, w`5, w`6, w`7, w`8, w`9, w`10, w`11, w`12, w`13, w`14 w`15)

D4 = (w"`1, w"`2, w"`3, w"`4, w"`5, w"`6, w"`7, w"`8, w"`9, w"`10, w"`11, w"`12)

D5 = (w""1, w""2, w""3, w""4, w""5, w""6, w""7, w""8, w""9, w""10,..., w""32, w""33, w""34)

LDA - це ітеративний процес

Перша ітерація LDA:

На першій ітерації LDA випадковим чином призначає теми кожному слову в документі. Теми позначаються літерою k. Отже, в нашому корпусі слова в документах будуть пов'язані з деякими випадковими темами, як показано нижче:

D1 = (w1 (k5), w2 (k3), w3 (k1), w4 (k2), w5 (k5), w6 (k4), w7 (k7), w8(k1))

D2 = (w`1(k2), w`2 (k4), w`3 (k2), w`4 (k1), w`5 (k2), w`6 (k1), w`7 (k5), w`8(k3), w`9 (k7), w`10(k1))

D3 = (w "1(k3), w "2 (k1), w "3 (k5), w "4 (k3), w "5 (k4), w "6(k1),..., w "13 (k1), w "14(k3), w "15 (k2))

D4 = (w"`1(k4), w"`2 (k5), w"`3 (k3), w"`4 (k6), w"`5 (k5), w"`6 (k3) ..., w"`10 (k3), w"`11 (k7), w"`12 (k1))

D5 = (w"`1 (k1), w"`2 (k7), w"`3 (k2), w"`4 (k8), w"`5 (k1), w"`6(k8) ..., w"`32(k3), w"`33(k6), w"`34 (k5))

На виході ми отримаємо Документи, що складаються з Тем і Теми, що складаються зі слів:

Документи є сумішшю тем:

D1 = k5 + k3 + k1 + k2 + k5 + k4 + k7 + k1

D2 = k2 + k4 + k2 + k1 + k5 + k2 + k1 + k5 + k3 + k7 + k1

D3 = k4 + k5 + k3 + k6 + k5 + k3 + ... + k3 + k7 + k1

D3 = k1 + k7 + k2 + k8 + k1 + k8 + ... + k3 + k6 + k5

Теми - це суміш слів:

K1 = w3 + w8 + w`4 + w`6 + w'10 + w "2 + w "6 + ... + w "13 + w"`12 + w""1 + w""5

K2 = w4 + w`1 + w`3 + w "15 + ... + w""3 + ...

K3 = w2 + w'8 + w "1 + w "4 + w "14 + w"`3 + w"`6 + ... + w"`10 + w""32 + ...

Аналогічно LDA видасть словосполучення для інших тем.

Після першої ітерації LDA надає початкові матриці документ-тема і тема-слово. Завдання полягає в оптимізації отриманих результатів, що LDA і робить, перебираючи всі документи і всі слова.

LDA робить ще одне припущення, що всі теми, які були визначені, є правильними, за винятком поточного слова. Отже, на основі цих вже правильних розподілів тема-слово, LDA намагається виправити і скоригувати розподіл теми для поточного слова за допомогою нового розподілу для якого:

LDA перебиратиме: кожен документ "D" і кожне слово "w

LDA обчислює дві ймовірності: p1 і p2 для кожної теми (k), де

P1: частка слів у документі (D), які наразі належать до теми (k)

P2: частка приналежності до теми (k) у всіх документах, які походять від цього слова w. Іншими словами, p2 - це частка тих документів, в яких слово (w) також приналежне до теми (k)

Формула для p1 і p2 така:

P1 = частка (тема k / документ D) , і

P2 = пропорція (слово w / тема k)

Тепер, використовуючи ці ймовірності p1 і p2, LDA оцінює нову ймовірність, яка є добутком (p1*p2), і через цю добуткову ймовірність LDA визначає нову тему, яка є найбільш релевантною темою для поточного слова.

Перепризначення слова 'w' документа 'D' новій темі 'k' через добуток ймовірностей p1 * p2

Тепер LDA виконується для великої кількості ітерацій для кроку вибору нової теми "k", поки не буде отримано стаціонарний стан. Точка збіжності LDA досягається там, де вона дає найбільш оптимізоване представлення матриці термінів документа і матриці слів теми.

На цьому робота і процес латентного розподілу Діріхле завершується.

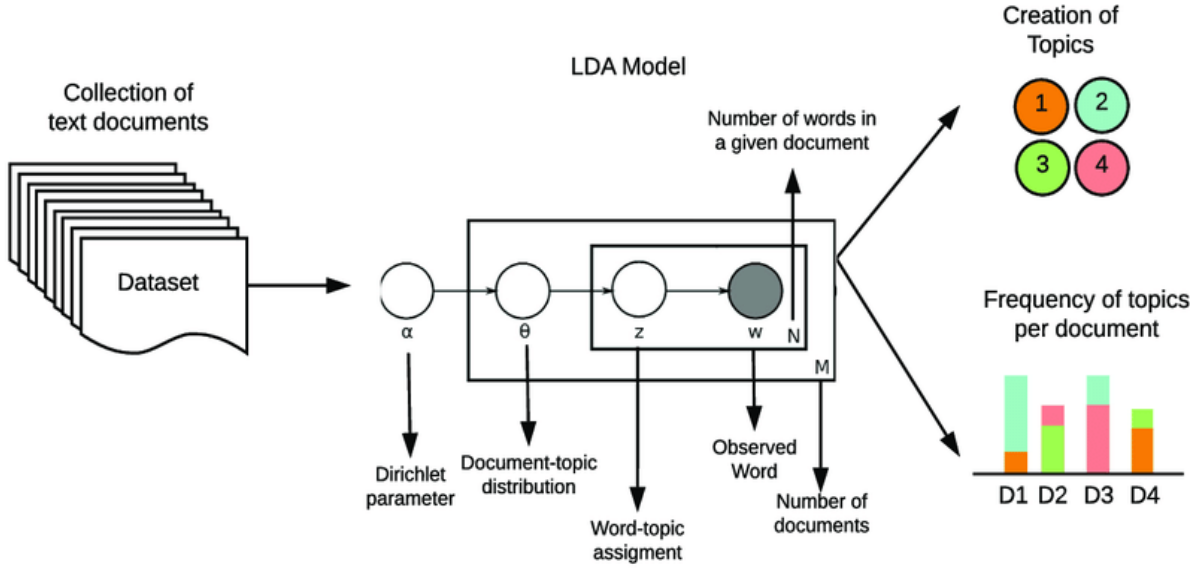


Рисунок 2.6.4: Наглядна схема роботи Latent Dirichlet Allocation.

2.7 Оцінка ефективності алгоритмів Latent Dirichlet Allocation і K-means у кластеризації колекцій текстових документів

Для оцінки ефективності обраних методів кластеризації ми реалізували їх з використанням мови програмування python, у середовищі pycharm. У ході розробки було використано наступні бібліотеки:

1. os: Бібліотека для взаємодії з операційною системою, яка надає функціональність для роботи з файловою системою, папками та шляхами до файлів.
2. numpy (NumPy): Бібліотека для наукових обчислень у Python. Вона надає підтримку для масивів великого розміру та математичних функцій для роботи з цими масивами.
3. sklearn (scikit-learn): Бібліотека для машинного навчання у Python. Вона містить реалізації різних алгоритмів машинного навчання, які можуть бути використані для класифікації, кластеризації, регресії та інших завдань.
4. sklearn.feature_extraction.text (TfidfVectorizer): Модуль scikit-learn, який надає інструменти для векторизації тексту на основі алгоритму TF-IDF (Term Frequency-Inverse Document Frequency).
5. sklearn.cluster (KMeans): Модуль scikit-learn, який містить реалізацію алгоритму K-means для кластеризації даних.
6. sklearn.decomposition (PCA): Модуль scikit-learn, який містить реалізацію методу головних компонент (PCA) для зменшення розмірності даних.
7. matplotlib.pyplot (plt): Підмодуль бібліотеки Matplotlib, який надає інструменти для візуалізації даних у вигляді графіків та діаграм.
8. seaborn (sns): Бібліотека для статистичної візуалізації даних у Python. Вона надає високорівневий інтерфейс для створення привабливих та інформативних графіків.
9. mpl_toolkits.mplot3d (Axes3D): Підмодуль бібліотеки Matplotlib, який надає інструменти для візуалізації тривимірних даних.

Ці бібліотеки широко використовуються для роботи з даними, машинного навчання та візуалізації у Python.

Scikit-learn (також відомий як sklearn) є однією з найпопулярніших бібліотек для машинного навчання та аналізу даних у Python. Вона надає широкий спектр алгоритмів та інструментів для класифікації, кластеризації, регресії, зменшення розмірності, підбору гіперпараметрів та багато іншого.

Деякі з головних особливостей sklearn включають:

1. Простота використання: sklearn має чистий та простий у використанні інтерфейс, який дозволяє легко створювати та налаштовувати моделі машинного навчання.

2. Широкий вибір алгоритмів: бібліотека містить реалізації багатьох класичних алгоритмів машинного навчання, таких як лінійна регресія, рішітка, метод опорних векторів (SVM), випадковий ліс, градієнтний бустинг і багато інших.
3. Інструменти для попередньої обробки даних: sklearn надає функціональність для обробки даних перед застосуванням моделей, включаючи шкалювання, кодування категоріальних змінних, заповнення пропущених значень, видалення викидів та інше.
4. Зручність узгодження моделей: sklearn має вбудовані інструменти для вибору найкращих гіперпараметрів моделі, такі як пошук по сітці (grid search) та перехресна перевірка (cross-validation).
5. Інтеграція з іншими бібліотеками: sklearn легко поєднується з іншими популярними бібліотеками Python, такими як NumPy та Pandas, що дозволяє зручно працювати з даними та виконувати різні операції.

Sklearn є інструментом для реалізації різних завдань машинного навчання та аналізу даних у Python, і вона надає зручний інтерфейс для швидкого прототипування та дослідження моделей.

Модуль sklearn.decomposition у бібліотеці scikit-learn надає інструменти для зменшення розмірності даних, зокрема метод головних компонентів (Principal Component Analysis, PCA).

PCA є одним із найпоширеніших методів зменшення розмірності. Він дозволяє знаходити нові нелінійні ортогональні ознаки, які найкраще пояснюють варіацію вхідних даних. Це досягається шляхом перетворення вихідних ознак у новий набір ознак, які називаються головними компонентами. Був використаний для створення візуалізації у поєднанні з matplotlib.pyplot (plt).

Для початку ми створили набір текстових документів для тестування і оцінки методів кластеризації. Було сформовано чотири тексових файли словники, із яких випадковим способом формувалися 570 текстових документів. Три словники частково перетинаються і складаються із термінів комп’ютерної тематики, у той час як четвертий складається із набору не пов’язаних з цією темою слів. Це було зроблено для того щоб оцінити як різні алгоритми впораються з такою задачею. На малюнку 2.7.1 наведено код для створення текстових документів:

```
import random
import string
import os

num_documents = 570
words_per_document = 50
output_folder = r'H:\1111\prr1\textfiles'
vocabulary_files = [
    r'H:\1111\prr1\voc1.txt',
    r'H:\1111\prr1\voc2.txt',
    r'H:\1111\prr1\voc3.txt',
    r'H:\1111\prr1\voc4.txt'
]

def generate_document(words_per_document, vocabulary):
    words = random.choices(vocabulary, k=words_per_document)
    document = ' '.join(words)
    return document

# Збираємо словник зі всіх файлів
vocabulary = []
for vocab_file in vocabulary_files:
    with open(vocab_file, 'r', encoding='utf-8') as file:
        vocabulary.extend(file.read().split())

# Генеруємо документи
documents = []
for _ in range(num_documents):
    vocab_file = random.choice(vocabulary_files)
    with open(vocab_file, 'r', encoding='utf-8') as file:
        vocabulary = file.read().split()
        document = generate_document(words_per_document,
vocabulary)
        documents.append(document)

# Зберігаємо документи у текстові файли
os.makedirs(output_folder, exist_ok=True)  # Створення папки, якщо
її немає
for i, document in enumerate(documents):
    filename = f'document_{i+1}.txt'
    filepath = os.path.join(output_folder, filename)
    with open(filepath, 'w') as file:
        file.write(document)
        print(f'Saved {filename} in {output_folder}')

print('All documents have been generated and saved.')
```

Рисунок 2.7.1 Створення колекції текстових докумннтів

Також було протестовано модуль векторизації(рисунок 2.7.2) і отримано корректні результати(рисунок2.7.3). Наведений код також збирає результати в один документ.

```
import os
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer,
TfidfTransformer

# Зчитування документів з текстових файлів
documents = []
input_folder = r'H:\1111\prr1\textfiles'
for filename in os.listdir(input_folder):
    filepath = os.path.join(input_folder, filename)
    with open(filepath, 'r') as file:
        document = file.read()
        documents.append(document)

# Векторизація документів
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(documents)

# Перетворення векторів у TF-IDF представлення
transformer = TfidfTransformer()
X_tfidf = transformer.fit_transform(X)

# Збереження результатів у файл
output_file = r'H:\1111\prr1\tfidf_results.txt'
with open(output_file, 'w') as file:
    # Записуємо кожен документ разом з його TF-IDF представленням
у файл
    for i, document in enumerate(documents):
        file.write(f"Document {i+1}:\n")
        file.write(f"{document}\n")
        file.write("TF-IDF:\n")
        for j, feature_name in
enumerate(vectorizer.get_feature_names()):
            tfidf_value = X_tfidf[i, j]
            file.write(f"{feature_name}: {tfidf_value}\n")
        file.write("\n")

print(f"TF-IDF results have been saved to {output_file}.")
```

Рисунок2.7.2 векторизація тексту.

Отримуємо представлення всіх документів у вигляді TF-IDF, частина розбору документу номер два представлена як приклад на рисунку 2.7.3. TF (term frequency - частота слова) - відношення числа входжень обраного слова до загальної кількості слів документа.

$$\mathrm{TF} = \frac{n_i}{\sum_k n_k},$$

Де n_i кількіть використання слова у дркументі, котра розділена на загальну кількість

IDF (inverse document frequency — обернена частота документа) — інверсія частоти, з якою слово зустрічається в документах колекції. Використання IDF зменшує вагу широковживаних слів[22].

$$\mathrm{IDF} = \log \frac{|D|}{|(d_i \supset t_i)|},$$

Де |D| — кількість документів колекції, а $|(d_i \supset t_i)|$ кількість документів, в яких зустрічається слово n_i.

```
Document 2:
cooldown game war rice cooldown america
TF-IDF:
accounting: 0.0
ai: 0.2056481126577558
algorithm: 0.0
america: 0.2749677851274686
animation: 0.0
architecture: 0.0
artificial: 0.0
automation: 0.0
backup: 0.0
big: 0.0
bluetooth: 0.0
browser: 0.0
cart: 0.1359580246111291
catrobot: 0.34370973140933575
cd: 0.1370987417718372
center: 0.0
cloud: 0.0
code: 0.0
computer: 0.0
cooldown: 0.27343213570206965
cybersecurity: 0.0
```

Рисунок 2.7.3 Матриця TF-IDF.

Наступним кроком була реалізація методу k-means з використанням зазначених на початку данного розділу методів. Код наведено у рисунку 2.7.4.

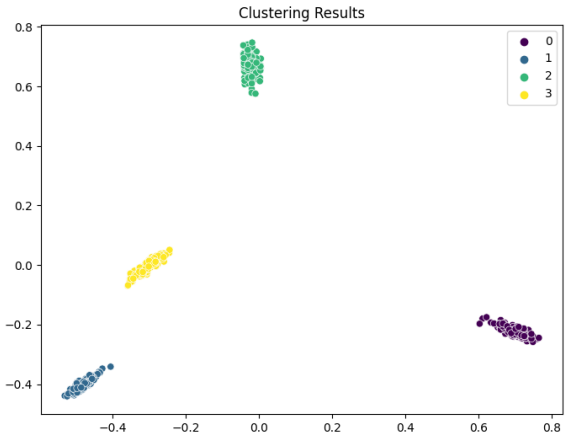
```
import os
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
import seaborn as sns
from mpl_toolkits.mplot3d import Axes3D

# Зчитування документів з текстових файлів
documents = []
input_folder = r'H:\1111\prr1\textfiles2'
for filename in os.listdir(input_folder):
    filepath = os.path.join(input_folder, filename)
    with open(filepath, 'r', encoding='utf-8') as file:
        document = file.read()
        documents.append(document)

# Векторизація документів у TF-IDF представлення
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(documents)

# Застосування алгоритму K-means для кластеризації
k = 5  # Кількість кластерів
kmeans = KMeans(n_clusters=k, random_state=42,tol=0.0001,
max_iter=1000)
kmeans.fit(X)

# Зменшення розмірності для візуалізації
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X.toarray())

# Візуалізація кластерів
colors = ['red', 'blue', 'green', 'orange', 'purple']
plt.figure(figsize=(8, 6))
for i in range(k):
    cluster_points = X_pca[kmeans.labels_ == i]
    plt.scatter(cluster_points[:, 0], cluster_points[:, 1],
c=colors[i], label=f'Cluster {i+1}')
plt.title('Clustering Results')
plt.legend()
plt.show()
# Візуалізація кластерів з використанням Seaborn
plt.figure(figsize=(8, 6))
sns.scatterplot(x=X_pca[:, 0], y=X_pca[:, 1], hue=kmeans.labels_,
palette='viridis')
plt.title('Clustering Results')
plt.show()
```

Рисунок 2.7.4 Реалізація алгоритму k-means мовою python.

```
# Застосування алгоритму PCA для зменшення розмірності до трьох
компонент
pca = PCA(n_components=3)
X_pca = pca.fit_transform(X.toarray())

# Візуалізація кластерів у тривимірному просторі
fig = plt.figure(figsize=(10, 8))
ax = fig.add_subplot(111, projection='3d')

for i in range(k):
    cluster_points = X_pca[kmeans.labels_ == i]
    ax.scatter(cluster_points[:, 0], cluster_points[:, 1],
cluster_points[:, 2], c=colors[i], label=f'Cluster {i+1}')

ax.set_title('Clustering Results')
ax.set_xlabel('Component 1')
ax.set_ylabel('Component 2')
ax.set_zlabel('Component 3')
ax.legend()
plt.show()

# Підрахунок кількості документів на кожен кластер
cluster_counts = np.bincount(kmeans.labels_)

# Візуалізація кількості документів у вигляді гістограми
plt.figure(figsize=(8, 6))
plt.bar(range(k), cluster_counts, color=colors)
plt.xlabel('Cluster')
plt.ylabel('Count')
plt.title('Document Counts per Cluster')
plt.show()
```

Рисунок 2.7.5 Реалізація алгоритму k-means мовою python, кінець.

Ми можемо легко змінювати кількість кластерів і маємо варіанти двовимірної і тривимірної візуалізації, розглянемо отримані результати на малюнках 2.7.6 – 2.7.11.



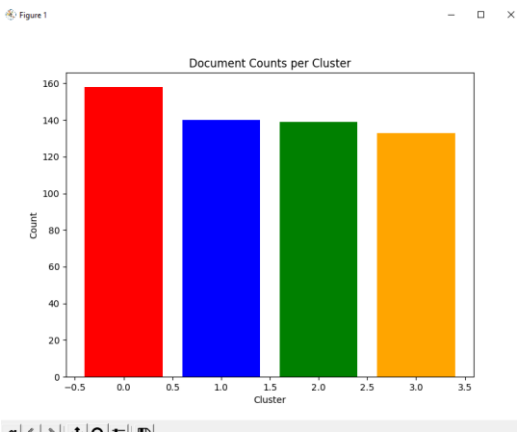


Рисунок 2.7.6 і рисунок 2.7.7 Результати при пошуку чьотирьох кластерів.

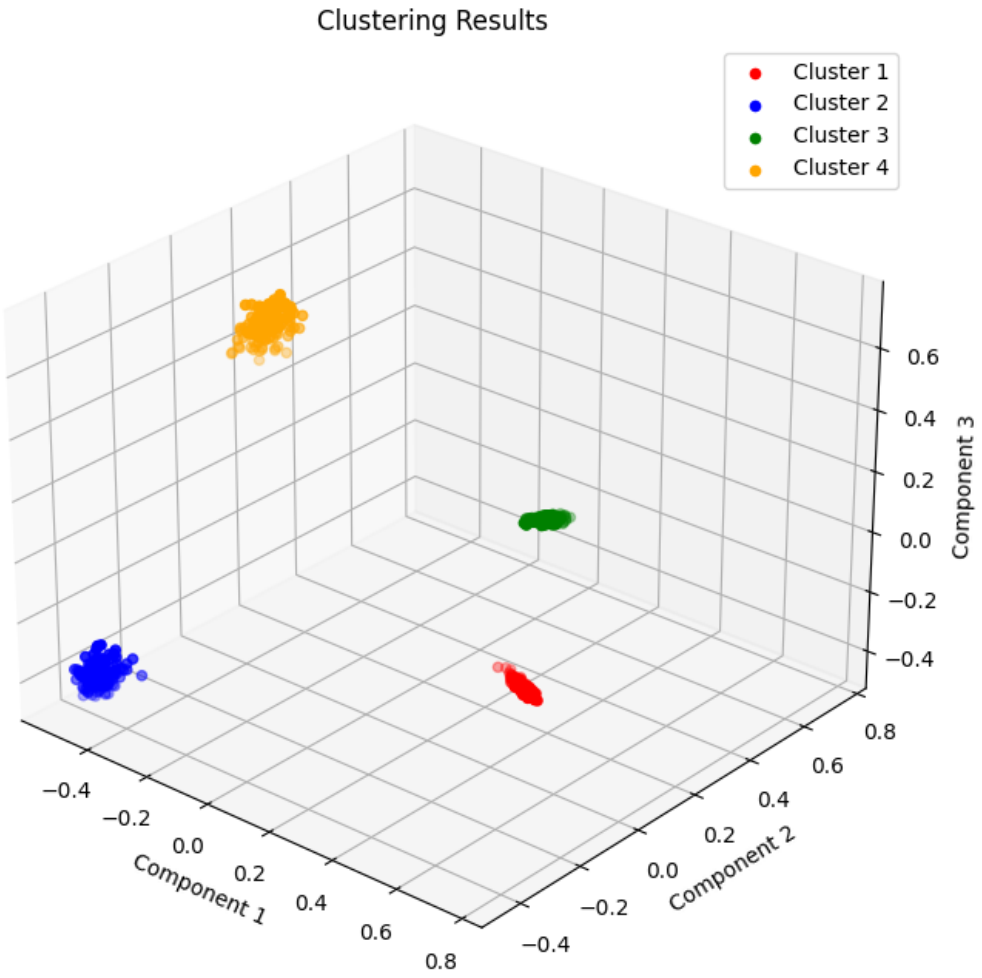


Рисунок 2.7.8 Результати при пошуку чьотирьох кластерів (3 виміри).

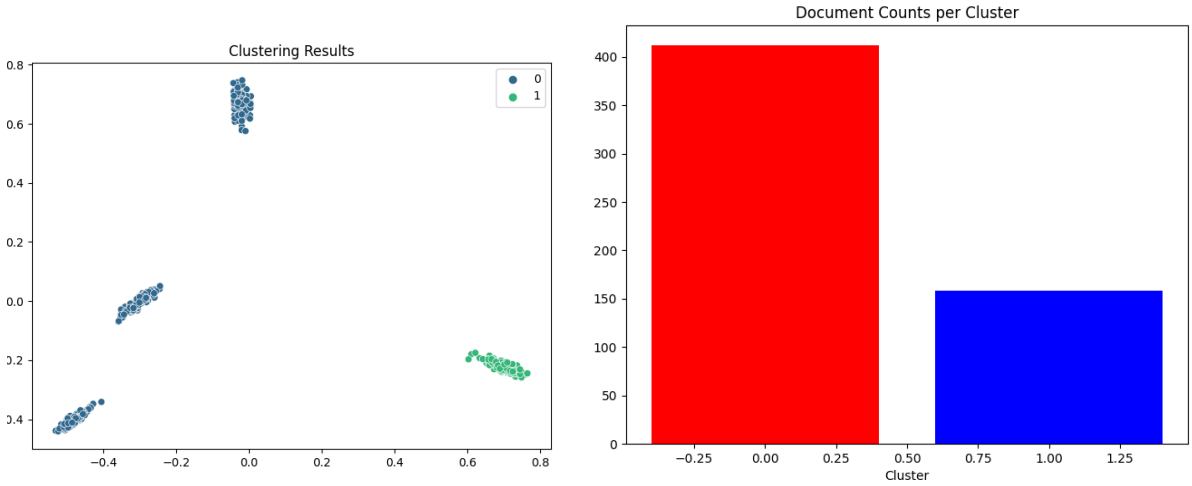


Рисунок 2.7.9 і рисунок 2.7.10 Результати при пошуку двох кластерів.

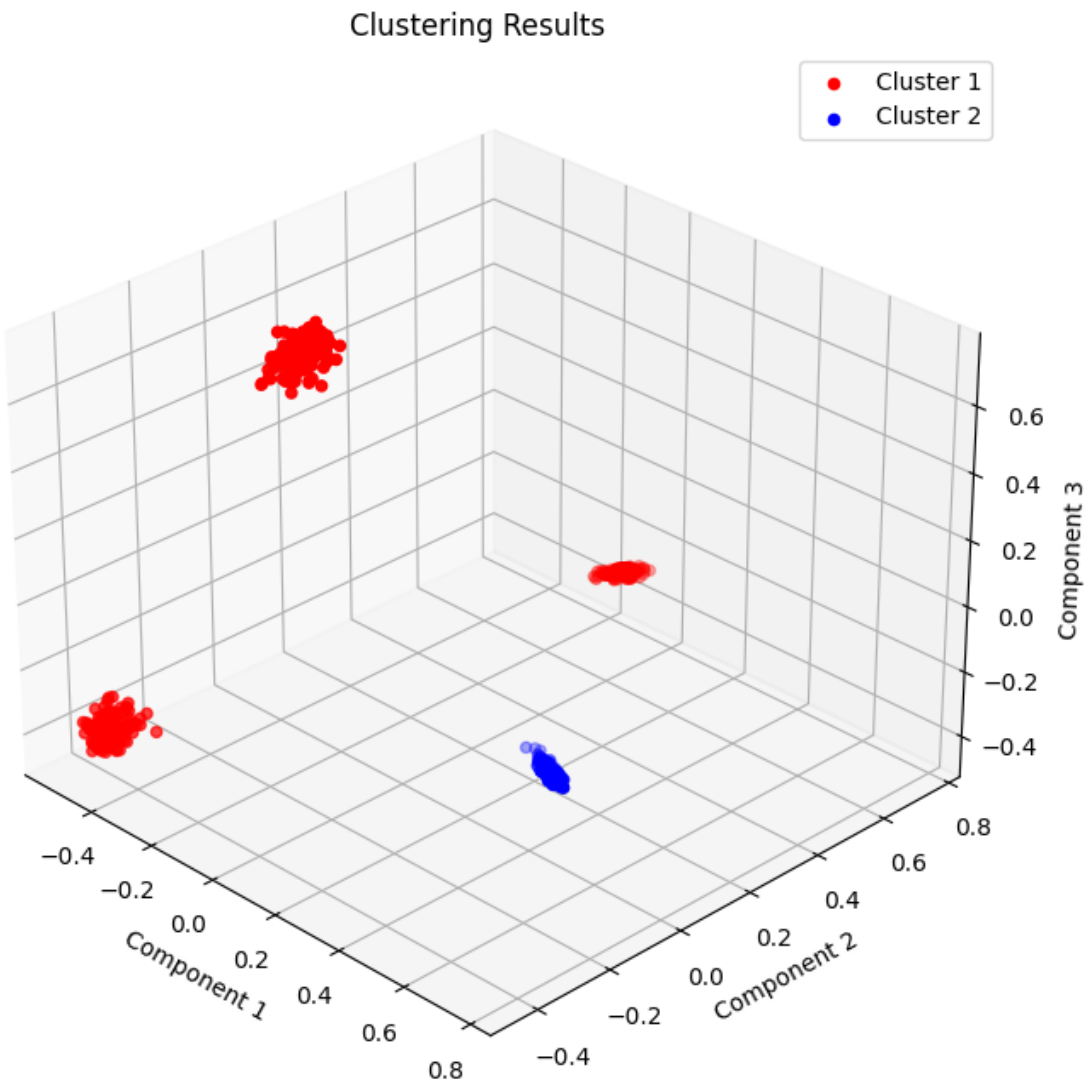


Рисунок 2.7.11 Результати при пошуку двох кластерів (3 виміри).

Як ми бачимо з малюнків 2.7.6-2.7.8 результатом будуть 4 подібних за розміром і структурою кластери. Але якщо ми задамо пошук двох, то алгоритм k-means зможі відділити тестовий не комп'ютерний набір документів через його віддалення по числовій оцінці у матриці TF-IDF і відповідно на отриманній візуалізації. Підтвердженням також є різниця більше ніж у два рази по кількосиі документів, що при початковій рівності кластерів по об'єму дає нам розуміння що за кластер вибраний другим. Важливо зазначити що у разі більшого наближення кластерів один до іншого, або їх більшого перетину метод к-середніх має високі шанси на повернення некорректрих результатів, або таких що дають мало корисних висновків.

Розглянемо нашу реалізацію алгоритму Latent Dirichlet Allocation або LDA.

На рисунках 2.7.12 і 2.7.3 подано код нашої реалізації данного алгорттму у середовищі pycharm з використаннями методів: TfidfVectorizer, numpy, sklearn.decomposition, matplotlib.pyplot, sklearn.manifold.

```
import os
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import LatentDirichletAllocation
import matplotlib.pyplot as plt
from sklearn.manifold import TSNE

# Зчитування документів з текстових файлів
documents = []
input_folder = r'H:\1111\prr1\textfiles2'
for filename in os.listdir(input_folder):
    filepath = os.path.join(input_folder, filename)
    with open(filepath, 'r', encoding='utf-8') as file:
        document = file.read()
        documents.append(document)

# Векторизація документів у TF-IDF представлення
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(documents)

# Побудова моделі LDA
num_topics = 13# Кількість тем
max_iterations = 1000  # Максимальна кількість ітерацій
convergence_tol = 0.001  # Параметр збіжності
lda_model = LatentDirichletAllocation(n_components=num_topics,
random_state=42)
lda_model.fit(X)

# Виведення отриманих тем
feature_names = vectorizer.get_feature_names()
for topic_id, topic in enumerate(lda_model.components_):
    top_words_indices = topic.argsort()[:-11:-1]
    top_words = [feature_names[i] for i in top_words_indices]
    print(f"Topic {topic_id + 1}:")
    print(", ".join(top_words))
    print()

# Приклад використання моделі для прогнозування теми нового
документа
new_document = "catrobot war america"
new_document_vector = vectorizer.transform([new_document])
new_document_topics = lda_model.transform(new_document_vector)
top_topic = np.argmax(new_document_topics)
print(f"Top topic for the new document: {top_topic + 1}")

# Отримання розподілу документів по темах
document_topics = lda_model.transform(X)
document_topic_distribution = np.sum(document_topics, axis=0) /
np.sum(document_topics)
```

Рисунок 2.7.12 Реалізація LDA 1 частина.

```
# Візуалізація розподілу документів по темах
plt.figure(figsize=(8, 6))
plt.bar(range(num_topics), document_topic_distribution,
align='center', color='blue')
plt.xlabel('Topic')
plt.ylabel('Document Proportion')
plt.title('Document Topic Distribution')
plt.xticks(range(num_topics), [f'Topic {i+1}' for i in
range(num_topics)])
plt.show()

# Отримання векторів тем для кожного документа
doc_topic_vectors = lda_model.transform(X)

# Застосування методу t-SNE для зменшення розмірності
tsne = TSNE(n_components=2, random_state=42)
tsne_vectors = tsne.fit_transform(doc_topic_vectors)

# Візуалізація результатів
plt.figure(figsize=(8, 6))
for i in range(num_topics):
    topic_points = tsne_vectors[np.argmax(doc_topic_vectors,
axis=1) == i]
    plt.scatter(topic_points[:, 0], topic_points[:, 1],
label=f'Topic {i+1}')
plt.title('t-SNE Visualization of LDA Topics')
plt.legend()
plt.show()
```

Рисунок 2.7.13 Реалізація LDA 2 частина.

Використовуючи таку ж колекцію текстових документів ми так само виконаємо кластеризацію і оцінемо результати, також буде отримано теми кластерів і перевірено можливість розподіляти нові документи.

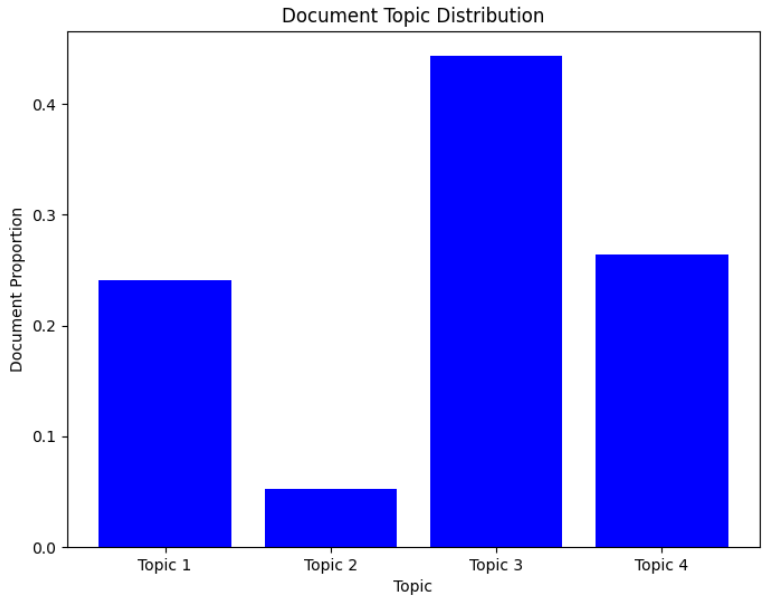


Рисунок 2.7.14 Гістограма розподілу документів по чотирьох темах-кластерах.

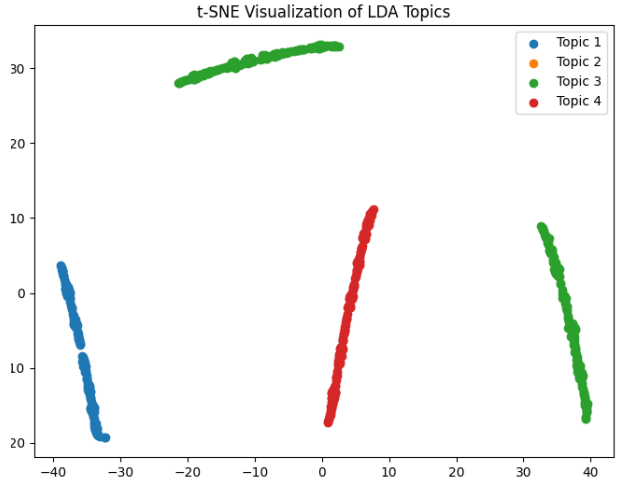


Рисунок 2.7.15 Зображення розподілу документів по чотирьох темах-кластерах.

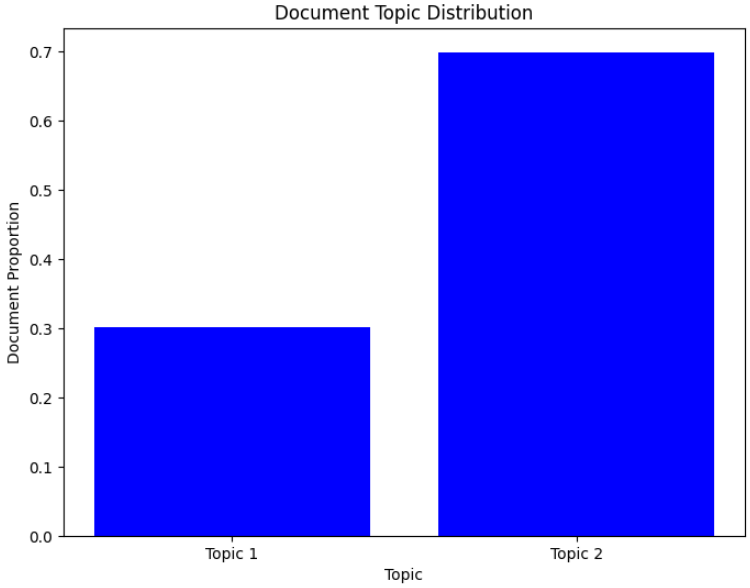


Рисунок 2.7.16 Гістограма розподілу документів по двох темах-кластерах.

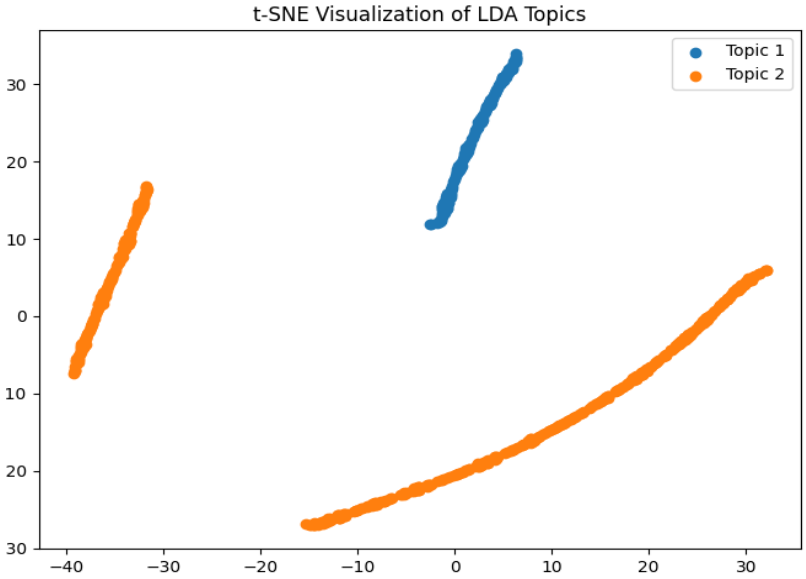


Рисунок 2.7.17 Зображення розподілу документів по двох темах-кластерах.

На відміну від попереднього, алгоритм скритого розподілу Діріхле оперує не тільки числовими коефіцієнтами, а і словами, завдяки чому може краще відшукувати приховані зв’язки між документами, крім того він позначає теми по яким розподіляв документи, що може значно полегшити роботу з данним алгоритмом. При обробці масива на 4 кластерииотримані наступні теми:

Topic 1:

repair, it, automation, engineering, user, storage, accounting, technician, literacy, testing

Topic 2:

firewall, data, cloud, cybersecurity, web, virus, photoshop, bluetooth, big, hdmi

Topic 3:

machine, hdmi, bluetooth, big, graphic, artificial, wi, fi, games, virtual

Topic 4:

fire, war, cooldown, catrobot, america, game, cart, cd, ai, rice

Top topic for the new document: 4

При розділі на дві такі:

Topic 1:

repair, it, automation, engineering, user, storage, accounting, technician, literacy, testing

Topic 2:

virus, fire, war, cooldown, catrobot, america, game, cart, cd, ai

Top topic for the new document: 2

Важливо зазначити що обидва рази він правильно відеіс новий документ до існуючою групи, що підтверджує його можливість працювати з новими документами без повторного навчання.

Як ми бачимо на рисунках 2.7.14 і 2.7.15 алгоритм LDA зафіксував що групи документів 2 і 3 мають немало повторних ключових слів, тому більшість документів направлена до третьої, в той час як перша їм оцінена близько до четвертої, хоча вона має ту ж тематику, так як має значно менше конкретрих термінів у собі, що перетиналися б з групами 2 в 3.

У разі пошуку двох кластерів алгоритм коректно розподіляю документи на відповідні групи, навіть якщо положення на двовимірній візуалізації не є очевидним, в цьому LDA значно кращий за k-means.

2.8 Оптимізація пошуку документів в мережі Інтернет з використанням алгоритмів кластеризації

У цьому розділі роботи ми спираючись на теоретичні відомості зазначені у роботі раніше і отримані у останньому розділі практичні результати спообуємо порівняти ефективність методів k-means і Latent Dirichlet Allocation.

Для початку важливо зазначити, що алгоритми K-Means і LDA (Latent Dirichlet Allocation) зазвичай використовуються в різних контекстах і мають різні цілі, тобто K-Means є алгоритмом кластеризації, який використовується для групування схожих об'єктів у відповідності до їх взаємної відстані, метою K-Means є мінімізація внутрішньокластерної дисперсії, тобто знаходження таких центроїдів кластерів, що мінімізують суму квадратів відстаней між точками кластера та їх центроїдами. В той час як LDA є алгоритмом машинного навчання, що використовується для тематичного моделювання тексту. Він допомагає виявити латентні (приховані) теми в наборі документів і визначити ймовірність присутності цих тем у кожному документі. Метою LDA є розкриття структури тематичних зв'язків у текстових даних і кластеризація документів за їх схожістю до тем. Але у данній роботі ми порівнюємо їх як алгоритми кластеризації колекції текстових документів.

Розглянеио результати роботи алгоритмів на колекції реальних документів. Ми маємо 200 документів невеликих за об’ємом і тих що мають навколокомп’ютерну тематику, почнемо з алгоритма k-means.

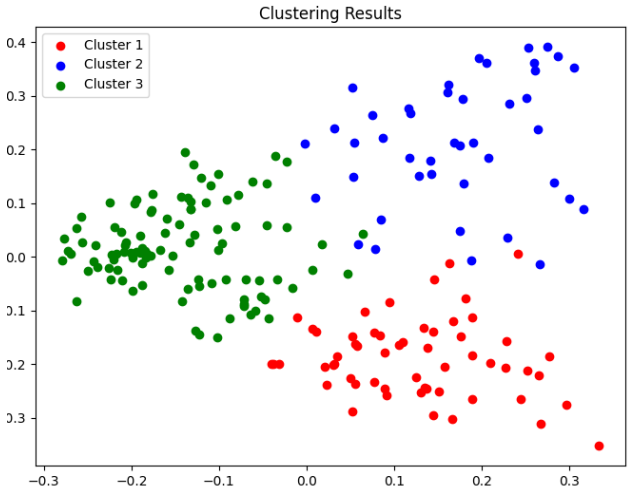


Рисунок 2.8.1 результат роботи k-means(3 кластери)

Як ми бачимо на рисунку 2.8.1 якщо ми вдало вказуємо кількість кластерів або знаємо її з початку в результаті ми отримаємо логічний розділ документів. В той час як при неправильному виборі кількості кластерів результат може бути значно менш зрозумілим, що млжна побачтти у додатку А де зібрано декілька результатів роботи опрацьованих алгоритмів. Через те що алгоритм працює виключно з числовими значеннями результат його робоии завжди потребує подальшого опрацювання.

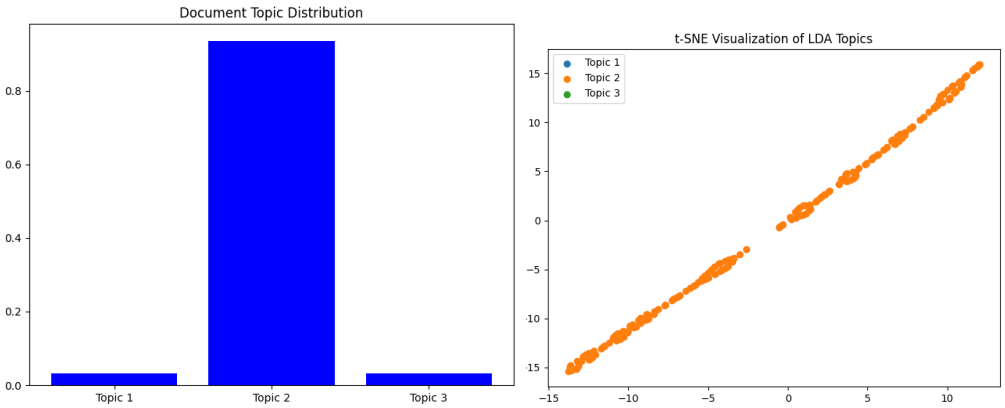


Рисунок 2.8.2 результат роботи LDA(3 кластери)

Також маємо такі ти теми:

Topic 1:
looping, buildrectpoint, printsubclass, sells, shells, topleft, toplefty, topleftx, she, string1
Topic 2:
the, to, of, and, in, you, is, java, that, class
Topic 3:
looping, buildrectpoint, printsubclass, sells, shells, topleftx, toplefty, topleft, she,

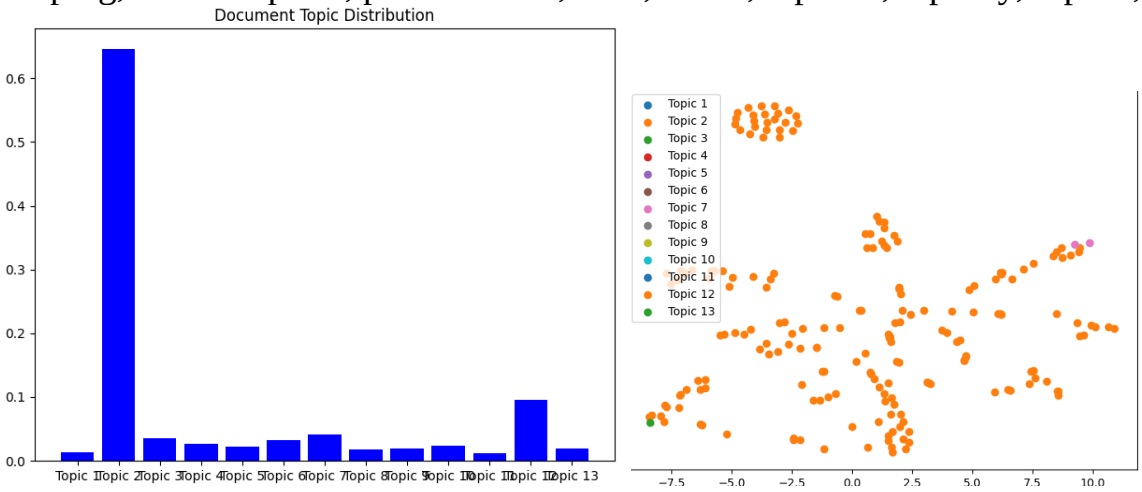


Рисунок 2.8.3 результат роботи LDA(13 кластерів)

На рисунках 2.8.2 і 2.8.3 ми бачимо результати роботи агоритму скритого розподілу Діріхле (LDA). У обох варіантах розподілу ми можемо простежити значне домінування однієї групи над інщими, 94 відсотки і 66 відсотків відповідно. Данний результат стає розумілим якщо ми розглянемо теми цих груп (повний список тем і разі розподілу на 13 кластерів поданий у додатку А):

1. the, to, of, and, in, you, is, java, that, class
2. the, to, of, and, in, you, is, java, that, class

Як ми бачимо вони ідентичні і складаються із технічних частин мови, а також слів ‘джава’ ти ‘клас’ які є загальною темою данної добірки документів. Маючи такий результат ми робимо висновок про надзвичайну важливість технічної підготовки документів до роботи із ними з використанням данного алгоритму. Також із повного сиску тем ми розуміємо логіку розподілу, наприклад кластери з переважаючим наповненням кодом, або тема 4 в документах з якої наявна інформація про техніку.

Аналізуючи отримані результати важливо зазначи і технічну складність алгоритму. Для методк к-середніх вона лінійна і дорівнює O(knT), де n – кількість документів, k – кількість кластерів, T – кількість ітерацій. В той час як для LDA вона повністю залежить від розміру документів і їх кількості, а так як із часом людство продукує геометрично більше інформації, а відповідно і текстових документів із якими працюють пошукові системи то інтеграція данного методу може визивати значні складності. Метод k-means не може опрацьовувати поступаючі нові документи, а тому сам по собі може бути використаний у обмеженому секторі задач і виходячи із отриманих експерементальних данних ми розуміємо що на данний момент він у більшій степені підходе на роль допоміжного інструменту, або частини багатоступеневого аналізу об’ємного кластеру документів. В ході роботи з данними алгоритмами і їх реалізацією було помітно різницю в скорості роботи методів з перевагою у першого, а з використанням способів оптимізації, наприклад k-means++ і роботі з дійсно великим об’ємом дпнних ця різниця тільки збільшиться.

Після тривалої роботи з цими методами сформувалося спостереження що метод к-середніх потребує не менше роботи після його завершення ніж при підготовці данних, в той час як LDA після підготовки файлів для його корректної роьоти, видає пов’язані тематично результати і добре розрізняє документи однієї теми, хоча і з варіативним набором лексем, від документів іншої теми. На данний момент ясним стає перевага алгоритму скритого розподілу Діріхле у контексті пошуку текстових документів в інтернеті або в об’ємній колекції інформації, в той час як алгоритм к-середніх з використанням методів оптимізації залишається оптимальним для роботи зі значними числовмми масивами, або у випадках коли значення швидкості пошуку значно доиінує над точністю та шириною отриманих результатів.

ВИСНОВКИ

Так як кількість інформації що продукує людство неспинно зростає було вирішено розглянути проблеми пов’язвні із цим, у ході кваліфікаційної роботи було проаналізовано різні методи кластеризації в контексті інформаційного пошуку, а саме використання їх для розподілу великих обсягів інформації по тематичним групам. Наявність легкого доступу до мережі Інтернет є неймовірно важливою і корисною частиною нашого життя, проте якісне використання данного невичерпного джерела інформації неможливе без добре налаштованих систем інформаційного пошуку, котрі в свою чергу неможливі без ефективної кластеризації масивів текстових документів – вмістилищ інформації.

Було вирішено розглянути докладно декілька розповсюджених методів кластеризацій, а саме: K-means (k-середніх), Hierarchical Clustering (ієрархічна кластеризація), Latent Dirichlet Allocation (LDA), DBSCAN (Density-Based Spatial Clustering of Applications with Noise), Mean Shift. Після вивчення їх принципів роботи, особливостей, плюсів і мінусів було обрано провести аналіз і порівняння двох з них: K-means як класичний математичний метод кластеризації і Latent Dirichlet Allocation як аналітичний метод кластеризації і пошуку скритих тематичних зв’язків між текстовими документами масиву.

З метою аналізу та порівняння процесу та результатів роботи цих алгоритмів було вирішено реалізувати іх зв допомогою мови програмування Python у середовищі Pycharm, з використанням ряду бібліотек як … для реалізації алгоритмів чи … для візуалізації отриманих результатів. Все це дозволило зробити висновки, що цілі та завдання КР успішно досягнуті, що також підтверджує практичну значущість проекту, а саме:

1. Досліджено сучасний інформаційний пошук;
2. Досліджено основні методи кластерізації;
3. Два обрані для дослідження методи були реалізовані;
4. Проведено ряд експериментів із використанням різних наборів данних;
5. Було порівняно обрані методи і зроблено висновки щодо викоримтання їх для оптимізації пошукуінформації в мережі Інтернет.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. “Techniques of cluster algorithms in data mining // Data Mining and Knowledge Discovery” Grabmeier J., Rudolph A. – October 2002. – Vol. 6, № 4. – p. 303-360.

2. “Survey of clustering algorithms // IEEE Transactions on Neural Networks.” – Xu R., Wunsch D. May 2005. – Vol. 16, № 3. – c. 645-678.

3. “Efficient phrase-based document indexing for web document clustering // IEEE Transactions on Knowledge and Data Engineering.” – Hammouda K.M., Kamel M.S October 2004. – Vol. 16, № 10. – c. 1279-1296.

4. “Web mining with relational clustering // International Journal of Approximate Reasoning.” – Runkler T.A., Bezdek J.C. February 2003. – Vol. 32, №. 2-3. – c. 217-236.

5. “A scalable hierarchical fuzzy clustering algorithm for text minin”– Rodrigues E.M., Sacks L. December 16-18, 2004. – c. 269-274.

6. https://uk.wikipedia.org/wiki/Ієрархічна_кластеризація

7. “Бізнес-аналітика багатовимірних процесів
мультимедійний навчальний посібник”- Т. С. Клебанова, Л. С. Гур’янова, Л. О. Чаговець, О. В. Панасенко, О. А. Сергієнко, Р. М. Яценко Харків, 2020

8. https://online.stat.psu.edu/stat505/lesson/14/14.4 (14.8)

9. Microsoft PowerPoint - Lect 11_ NON HIERARCHICAL CA (unife.it)
“Statistics for Economics and Business” Stefano Bonnini & Valentina Mini non hierarchical methods Lecture 11 – 27nd

10. Кластерний аналіз — Вікіпедія (wikipedia.org)

11. “Latent dirichlet allocation” - David M. Blei, Andrew Y. N,Michael I. Jordan The Journal of Machine Learning ResearchVolume 3pp 993–1022 01 March 2003

12. https://en.wikipedia.org/wiki/Hierarchical_clustering

13. https://en.wikipedia.org/wiki/DBSCAN

14. " МАШИННЕ НАВЧАННЯ: МЕТОДИ ТА МОДЕЛІ "- К. Ю. Кононова Харків 2020

15. https://uk.wikipedia.org/wiki/ Кластеризація_методом_к–середніх

16. “Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey.”- Jelodar, H., Wang, Y., Yuan, C. Multimed Tools Appl 78, 15169–15211 (2019).

17. https://en.wikipedia.org/wiki/Plate_notation

18.https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation#:~:text=In%20natural%20language%20processing%2C%20Latent,of%20a%20Bayesian%20topic%20model.

19. https://www.analyticsvidhya.com/blog/2021/06/part-2-topic-modeling-and-latent-dirichlet-allocation-lda-using-gensim-and-sklearn/

20. “МЕТОДИ ТА ТЕХНОЛОГІЇ ОБЧИСЛЮВАЛЬНОГО ІНТЕЛЕКТУ”- І. В. ФЕДОРІН Київ КПІ ім. Ігоря Сікорського 2022

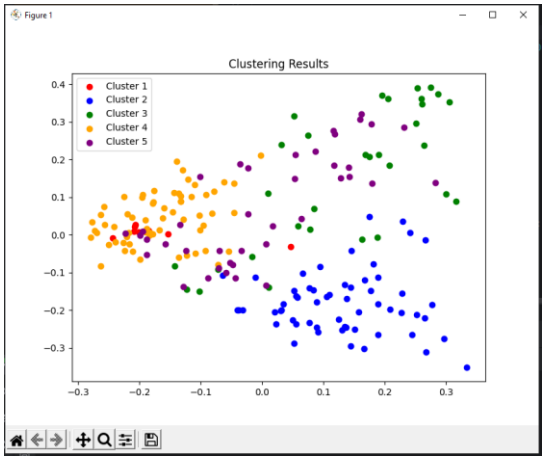
21. Graphical-model-of-latent-Dirichlet-allocation-LDA.png (850×313) (researchgate.net)

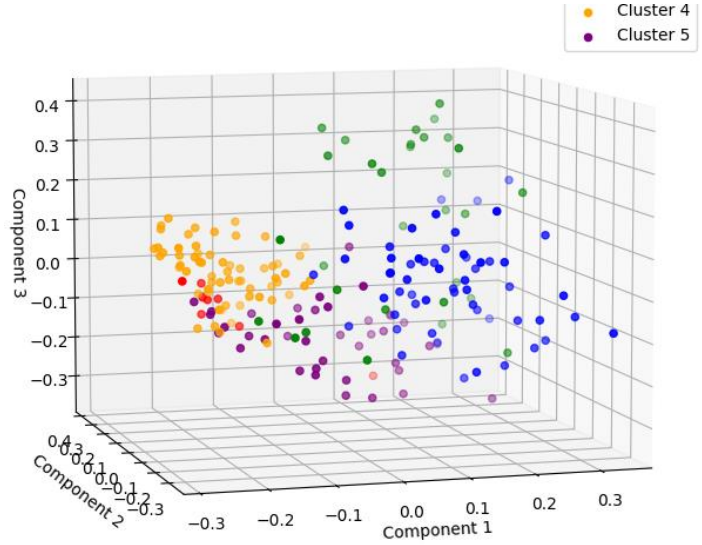
22. https://uk.wikipedia.org/wiki/TF-IDF

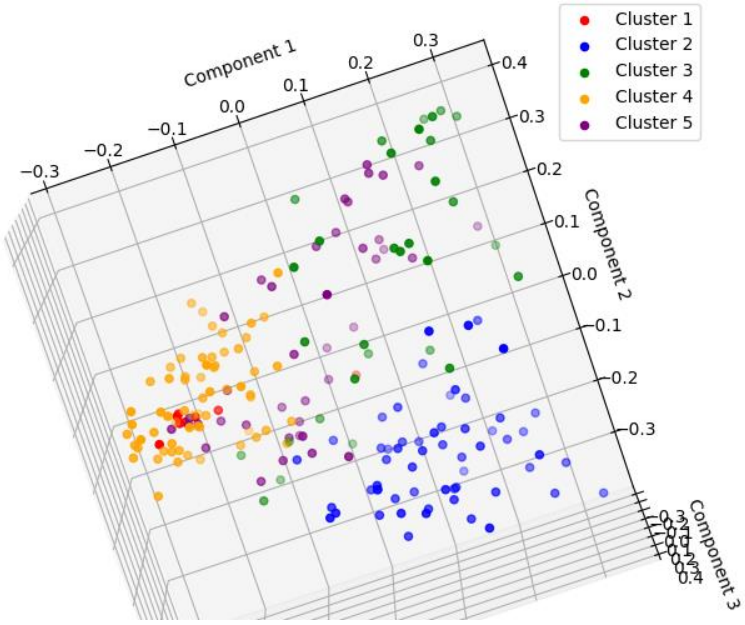
ДОДАТКИ

Додаток А

Результат роботи k-means(5 кластерів)







Результат роботи LDA(13 кластерів)

Topic 1:
module, apple, greenapple, shift, average, overflow, oop, surface, argsi, state
Topic 2:
the, to, of, and, in, you, is, java, that, class
Topic 3:
thread, space, thisname, userinput, screen, year, aligns, investment, morehelloapplet, comment
Topic 4:
interrupt, vehicle, directory, reads, signal, car, winter, play, yamaha, 20
Topic 5:
passbyreference, conditional, onetozero, label, javac, rectangles, underscore, helloworldclass, shortcut, layout
Topic 6:
date, buildrect, printsubclass, bytecode, interpreter, looping, 1010, printclass, round, compiled
Topic 7:
casting, logical, precedence, covered, instanceof, division, parentheses, level, edition, cover
Topic 8:
sea, str2, str1, rangeclass, sells, shells, makerange, myrect2, alpha, she
Topic 9:
indentation, screen, clone, 16, helloworld, 18, sites, str1, str2, showatts
Topic 10:
interpreter, bitwise, bytecode, virtual, gdrawstring, paintgraphics, null, years, codebase, pc
Topic 11:
bits, systemin, comments, referring, unary, 46, nongui, char, 64, concatenate
Topic 12:
array, method, systemoutprintln, loop, arguments, listing, motorcycle, engine, line, constructor
Topic 13:
date, namedpoint, specifier, element, concatenation, heard, bodyofloop, house, printme, aprogram

Додаток Б

Відомості матеріалів дипломної роботи

		Позначення	Найменування	Кількість	Примітка
1					
2			Документація		
3					
4		САУ.КР.23.15.ПЗ	Пояснювальна записка	89	Формат А4
5					
6		САУ.КР.23.15.ДМ	Демонстраційні матеріали	12	А4
7					
8		САУ.КР.23.15.КР	Копія роботи	1	Диск CD-R

					САУ.КР.23.15.ДА.ПЗ				
Зм	Лист	№ докум.	Підпис	Дата					
Розроб.		Рублевський			**Матеріали кваліфікаційної роботи**	Літ		Аркуш	Аркушів
К. розд.		Одновол							
Керівн.		Одновол				НТУ «ДП», 12; 124-19-1			
Н. контр.		Хом`як							
Зав. каф.		Желдак							

Додаток В

Рецензія

на кваліфікаційну роботу бакалавра

студентки групи *124 – 19 – 1 Рублевський І.О.*

спеціальності 124 Системний аналіз

Тема кваліфікаційної роботи:

Розробка математичної моделі оптимізації пошуку інформації в мережі інтернет з використанням методів кластеризації.

Обсяг кваліфікаційної роботи: _____

Висновок про відповідність кваліфікаційної роботи завданню та освітньопрофесійній програмі спеціальності _____

_____

Загальна характеристика кваліфікаційної роботи, ступінь використання нормативно-методичної літератури та передового досвіду _____

_____

Позитивні сторони кваліфікаційної роботи: _____

_____

Основні недоліки кваліфікаційної роботи:_____

_____

Кваліфікаційна робота в цілому заслуговує оцінки: _____

З урахуванням висловлених зауважень автор заслуговує присвоєння освітньої кваліфікації «бакалавр з системного аналізу».

Рецензент,

науковий ступінь, вчене звання, посада _____ / ПІБ

Додаток Г

Відгук
на кваліфікаційну роботу бакалавра
студента групи 124 – 19 – 1
спеціальності 124 Системний аналіз

Тема кваліфікаційної роботи: Розробка математичної моделі оптимізації пошуку інформації в мережі інтернет з використанням методів кластеризації.
Обсяг кваліфікаційної роботи __90__ стор.
Мета кваліфікаційної роботи: оптимізація процесу пошуку текстових документів в Інтернеті шляхом реалізації математичних моделей, які базуються на методах кластеризації.
Актуальність теми _полягає у тому, що при пришвидшенні зростання кількості інформації дуже важливо оперативне її засвоєння, і данна кваліфікаційна_____ _робота побудована навколо аналізу кластерних методів оптимізації взаємодії з наявною і новою інформацією._____ __
Тема кваліфікаційної роботи безпосередньо пов'язана з об'єктом діяльності бакалавра спеціальності 124 Системний аналіз, оскільки вона повністю базується на методах кластерного аналізу та на оцінці їх ефективності. _
Виконані в кваліфікаційній роботі завдання відповідають вимогам ступеня бакалавра.
Оригінальність наукових рішень полягає в проведенні наочних експерементів із методами кластерного аналізу, їх дослідженні та порівнянні._____
Практичне значення результатів кваліфікаційної роботи полягає в виборі оптимального підходу до інформаційного пошуку текстових документів в мережі інтернет, використовуючи методи кластерного розподілу._____
Висновки підтверджують можливість використання результатів роботи в створенні інформаційно пошукових систем для сайтів чи баз данних компаній._
Оформлення пояснювальної записки та демонстраційного матеріалу до неї виконано згідно з вимогами. Роботу виконано самостійно, відповідно до завдання та у повному обсязі (в разі невідповідності – вказати)
У роботі відзначено такі недоліки: _____
_____
_____
Кваліфікаційна робота в цілому заслуговує оцінки:_____

З урахуванням висловлених зауважень автор заслуговує присвоєння освітньої кваліфікації «бакалавр з системного аналізу».

Керівник кваліфікаційної роботи бакалавра,
науковий ступінь, вчене звання, посада _____ / ПІБ