Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

ДОСЛІДЖЕННЯ ТА РЕАЛІЗАЦІЯ МЕТОДУ СЕМАНТИЧНОЇ КЛАСТЕРИЗАЦІЇ НОВИН ВІДПОВІДНО ДО ЇХ ТЕМАТИКИ
(тема)

Виконав:
студент 2 курсу, групи ІНФм-23-1
Гончарова О.В.
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Кобилін О.А.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Кобилін О.А.
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Гончаровій Оксані Володимирівні
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та реалізація методу семантичної кластеризації новин відповідно до їх тематики

затверджена наказом по університету від 25 листопада 2024 року № 1246Ст

2. Термін подання студентом роботи до екзаменаційної комісії 24 грудня 2024 р.

3. Вихідні дані до роботи математичні моделі семантичної кластеризації текстових даних, перелік використовуваних програмних засобів: теоретичні відомості про методи семантичної кластеризації, мова програмування Python.

4. Перелік питань, що потрібно опрацювати в роботі

1. Аналіз особливостей застосування кластеризації в областях пов'язаних з обробкою тексту

2. Дослідження існуючих методів семантичної кластеризації текстових даних

3. Аналіз роботи кожного з методів при однакових вхідних даних на прикладі реальних або симульованих даних

4. Надання оцінки отриманих результатів та їх візуалізація

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність проблеми семантичної кластеризації текстових даних, постановка задачі, схеми роботи алгоритмів.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	25.11.2024	
2	Аналіз завдання, підбір літератури	25.11.24-27.11.24	
3	Аналіз літератури з досліджуваної проблеми	27.11.24-28.11.24	
4	Аналіз технічних засобів	28.11.24-29.11.24	
5	Розробка методу	29.11.24-30.11.24	
6	Програмна реалізація	30.11.24-03.12.24	
7	Оформлення пояснювальної записки	03.12.24-05.12.24	
8	Перевірка на плагіат	09.12.2024	
9	Рецензування	16.12.2024	
10	Підготовка презентації та доповіді	18.12.2024	
11	Занесення роботи в електронний архів	02.01.2025	
12	Попередній захист кваліфікаційної роботи	02.01.2025	

Дата видачі завдання 25 листопада 2024 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Кобилін О.А.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 71 с., 2 табл., 25 рис., 65 джерел.

СЕМАНТИЧНА КЛАСТЕРИЗАЦІЯ, K-MEANS, LDA, LSA, DBSCAN, SPECTRAL CLUSTERNG, ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ТЕКСТОВИХ ДАНИХ, ПОПЕРЕДНЯ ОБРОБКА ТЕКСТУ.

Об’єктом дослідження є методи семантичної кластеризації.

Метою дослідження є порівняння методів семантичної кластеризації з метою їхнього застосування для кластеризації новин відповідно до їх тематики.

Проведено аналіз та порівняння різних підходів до семантичної кластеризації, враховуючи їхню ефективність та пристосованість до конкретної задачі обробки новин. Проведено експериментальне дослідження на основі реальної вибірки даних з новинами для оцінки якості та продуктивності програмно розроблених методів.

У результаті дослідження отримано якісні та швидкісні характеристики роботи методів K-Means, LDA, LSA, Spectral Clustering та DBSCAN з реальною вибіркою новинних даних.

SEMANTIC CLUSTERING, K-MEANS, LDA, LSA, DBSCAN, SPECTRAL CLUSTERING, TEXT MINING, TEXT PREPROCESSING.

The object of the research is the methods of semantic clustering.

The aim of the research is to investigate and compare semantic clustering methods for their application in clustering news according to their topics.

An analysis and comparison of different approaches to semantic clustering have been conducted, considering their effectiveness and adaptability to the specific task of news processing. Experimental research was carried out on a real dataset of news to evaluate the quality and performance of the implemented methods.

As a result of the research, the qualitative and performance characteristics of the K-Means, LDA, LSA, Spectral Clustering, and DBSCAN methods were obtained on a real news dataset.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів 7
Вступ 8
1 Аналіз проблематики семантичної кластеризації новин відповідно до їх тематики 9
1.1 Актуальність проблеми інтелектуального аналізу новин 9
1.2 Особливості застосування кластеризації в областях пов’язаних з обробкою тексту 11
1.2.1 Загальний алгоритм кластеризації текстових даних 12
1.2.2 Особливості попередньої обробки текстових даних 14
1.2.3 Особливості побудови моделей семантичної репрезентації текстових даних……...……16
1.2.4 Міри семантичної подібності 17
1.3 Використання семантичної кластеризації в сфері новин 20
1.3.1 Проблематика врахування семантичних зв’язків у новинах……..20
1.3.2 Перспективи розвитку семантичної кластеризації новин 22
1.4 Постановка задачі дослідження 22
2 Математичні моделі семантичної кластеризації текстових даних 24
2.1 Базовий алгоритм K-Means 24
2.1.1 Модифікації алгоритму: Bisecting K-Means та Fuzzy C-Means 26
2.1.2 Переваги та недоліки K-Means та його модифікацій 29
2.2 Алгоритми ієрархічної кластеризації. Їх переваги та недоліки 30
2.3 Прихований розподіл Діріхле. Переваги та недоліки 32
2.4 Особливості LSI/LSA 35
2.5 Спектральна кластеризація 37
2.6 Алгоритм DBSCAN. Його особливості, переваги та недоліки 39

2.7 Особливості, переваги та недоліки кластеризації на основі векторних представлень слів 42
3 Дослідження методів семантичної кластеризації новин відповідно до їх тематики 44
3.1 Опис методології проведення експериментального порівняльного аналізу 44
3.2 Попередня обробка даних 47
3.2.1 Опис робочого датасету 47
3.2.2 Попередня обробка даних 47
3.2.3 Результати попередньої обробки датасету 48
3.3 Програмна реалізація та візуалізація результатів виконання методів семантичної кластеризації 49
3.3.1 Метод K-Means 49
3.3.2 Метод LDA 52
3.3.3 Метод LSA 54
3.3.4 Метод Spectral Clustering 57
3.3.5 Метод DBSCAN 59
3.4 Порівняння отриманих результатів 61
Висновки 63
Перелік джерел посилання 65

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ARI – Adjusted Rand Index (скоригований індекс Ренда)

BERT – Bidirectional Encoder Representations from Transformers (двоспрямовані кодувальні представлення з трансформерів)

DBSCAN – Density-Based Spatial Clustering Of Applications With Noise (кластеризація просторових даних на основі щільності з використанням шуму)

GES – Generalized Edit Similarity (узагальнена метрика редагування)

HAC – Agglomerative Hierarchical Clustering (агломеративна ієрархічна кластеризація)

HDC – Divisive Hierarchical Clustering (розділювальна ієрархічна кластеризація)

LDA – Latent Dirichlet Allocation (прихований розподіл Діріхле)

LSA – Latent Semantic Analysis (латентно-семантичний аналіз)

LSI – Latent Semantic Indexing (латентно-семантична індексація)

NLP – Natural Language Processing (обробка природної мови)

SVD – Singular Value Decomposition (декомпозиція сингулярного значення)

ВСТУП

У сучасному інформаційному суспільстві обробка та аналіз великого обсягу новинних матеріалів має велике значення для різноманітних сфер, включаючи медіа, політику, бізнес та науку. Великий обсяг новинних даних, що постійно зростає, ускладнює завдання їхнього оброблення та аналізу. У зв'язку з цим користувачі часто стикаються з труднощами пошуку релевантних матеріалів серед величезної кількості новинних джерел. Подання новин у відповідних категоріях та тематичних групах стає дедалі важливішим завданням для забезпечення користувачам швидкого та ефективного доступу до інформації, яка їх цікавить.

Проблематика кластеризації новинних потоків полягає у складнощах правильного визначення тематики новин через неоднозначність природної мови, багатозначність слів та контексту. Багато сучасних алгоритмів кластеризації не завжди здатні точно ідентифікувати семантичну близькість між новинами, що призводить до хибної класифікації або надмірної генералізації. Тому актуальним є пошук та дослідження методів, які могли б підвищити точність аналізу змісту новинних матеріалів, враховуючи різні аспекти природної мови та контексту.

Одним із ефективних підходів для розуміння та впорядкування новинних матеріалів є семантична кластеризація, яка забезпечує можливість систематизувати контент за змістовими критеріями, що сприяє створенню більш інтелектуальних і зручних для користувача інформаційних платформ.

Актуальність дослідження полягає у дослідженні та пошуку методів кластеризації для підвищення точності аналізу новинних матеріалів, враховуючи контекст.

1 АНАЛІЗ ПРОБЛЕМАТИКИ СЕМАНТИЧНОЇ КЛАСТЕРИЗАЦІЇ НОВИН ВІДПОВІДНО ДО ЇХ ТЕМАТИКИ

Кластеризація – це процес розподілу, або групування набору схожих об’єктів даних на підмножини разом у відповідності з певними критеріями схожості чи віддаленості. Отже, кластер – це набір об’єктів, схожих на кожен інший, але несхожих на об’єкти з інших кластерів. Це широко використовувана техніка в інтелектуальному аналізі даних, пошуку інформації та виявленні знань для пошуку прихованих шаблонів або об’єктів у даних іншої категорії [1, 2].

Процес кластеризації ж тексту стосується групування неструктурованої колекції документів у семантично пов’язані групи. У традиційних методах кластеризації документів документ розглядається як пакет слів, однак семантичне значення слова не враховується. Таким чином, більш інформативні характеристики, такі як вага концепції, важливі для досягнення точної кластеризації документів, і цього можна досягти за допомогою семантичної кластеризації документів, оскільки вона враховує більш значущі зв’язки [3, 4].

1.1 Актуальність проблеми інтелектуального аналізу новин

За останні кілька років інформаційний обсяг у мережі виріс вкрай швидко, і ця тенденція продовжується. З кожним днем ми стикаємося з все більшою кількістю текстових даних у всіх сферах нашого життя – від новин та статей до електронного листування та соціальних мереж. Цей надзвичайно великий обсяг даних потребує ефективних інструментів для аналізу, з метою виділення корисної інформації, виявлення загальних тенденцій та глибшого розуміння контексту [5-7].

Актуальність інтелектуального аналізу текстів на сьогоднішній день надзвичайно висока через ряд ключових факторів:

− зростання обсягу текстових даних. Інтернет та цифрові медіа забезпечують безпрецедентний доступ до величезного обсягу текстових даних, таких як веб-сторінки, соціальні медіа, електронні листи, наукові статті та інші джерела. Збільшення обсягу даних ставить завданням створення ефективних методів аналізу та витягнення інформації з цього потоку даних;

− потреба в автоматизації процесів. В умовах великого обсягу даних та швидкого темпу їхнього збільшення виникає потреба в автоматизованих інструментах для обробки та аналізу текстів. Інтелектуальний аналіз текстів дозволяє створювати автоматизовані системи для кластеризації, класифікації, екстракції інформації та інших завдань аналізу тексту;

− розвиток штучного інтелекту та обробки природної мови. Завдяки проривам у галузі штучного інтелекту та обробки природної мови (Natural Language Processing, NLP), стає можливим застосування новітніх методів та технік для інтелектуального аналізу текстів. Розвиток алгоритмів машинного навчання, глибокого навчання та нейронних мереж сприяє покращенню точності та ефективності аналізу [7].

Отже, у сучасному цифровому світі, заповненому величезним обсягом інформації, роль інтелектуального аналізу текстів стає надзвичайно важливою. Інформаційний вибух, спричинений зростанням мережі Інтернету та соціальних медіа, породжує неабияку потребу в здатності швидко та ефективно аналізувати текстові дані. Інтелектуальний аналіз текстів, який поєднує в собі методи штучного інтелекту, машинного навчання та обробки природної мови, надає засоби для вирішення цієї проблеми [6, 8].

Особливо важливою є ця проблема в контексті новин на сьогоднішній день: незаконні військові дії в Україні, сучасні вибори, глобальні кризи, такі як пандемія COVID-19, та інші події, які можуть стати об’єктом політичної маніпуляції або масової дезінформації. Дуже важливо вміти розрізняти достовірні новини та фейкові, або новини, спотвореним контентом. Тому

розвиток і вдосконалення методів інтелектуального аналізу новин залишається актуальним завданням для забезпечення інформаційної безпеки та свободи в сучасному світі.

1.2 Особливості застосування кластеризації в областях пов’язаних з обробкою тексту

Подібно до класифікації текстів, кластеризація текстів також є технологією обробки великої кількості текстів та їх поділу. Основна відмінність полягає в тому, що аналіз кластеризації текстів надає оптимальне поділ на категорії без необхідності попереднього ручного маркування категорій деяких документів [9, 10].

Кластеризація текстових документів спрямована на автоматичний поділ на групи за подібністю вмісту. Кожна група (або кластер) складається з текстових документів, які схожі всередині групи (мають високу внутрішньо кластерну схожість) і несхожі на документи інших груп (мають низьку міжкластерну подібність) (рис. 1.1).

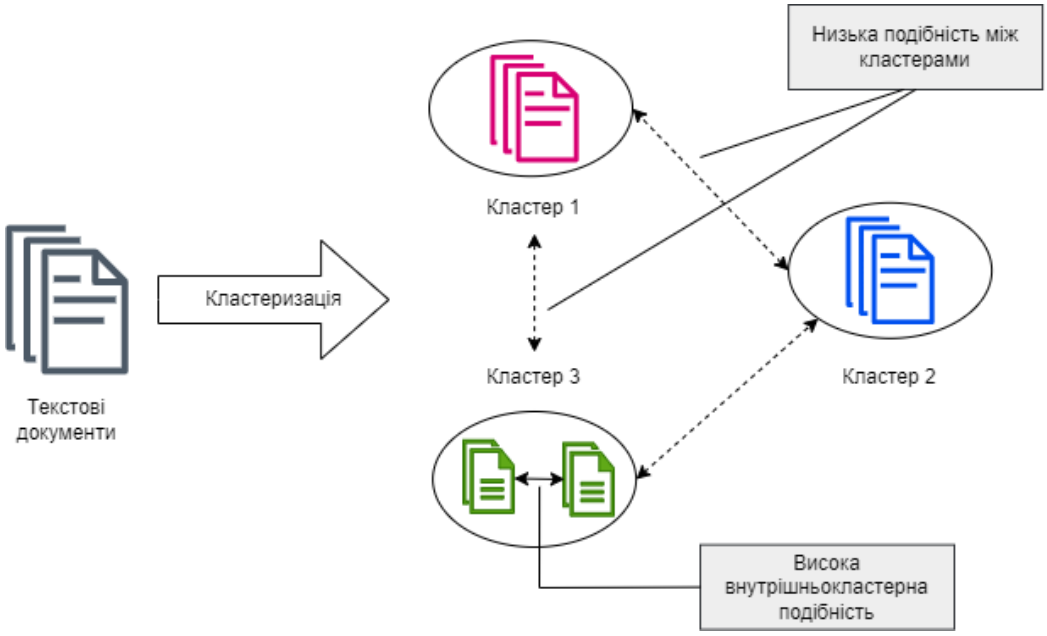

Рисунок 1.1 – Кластеризація текстових документів

1.2.1 Загальний алгоритм кластеризації текстових даних

Кластеризація текстів є ненавчальним процесом, який не залежить від попередніх знань про збір даних, і ґрунтується виключно на відносинах подібності між документами у колекції для поділу документів на кілька кластерів. Загальний математичний опис кластеризації текстів може бути зображений наступним чином (рис. 1.2).

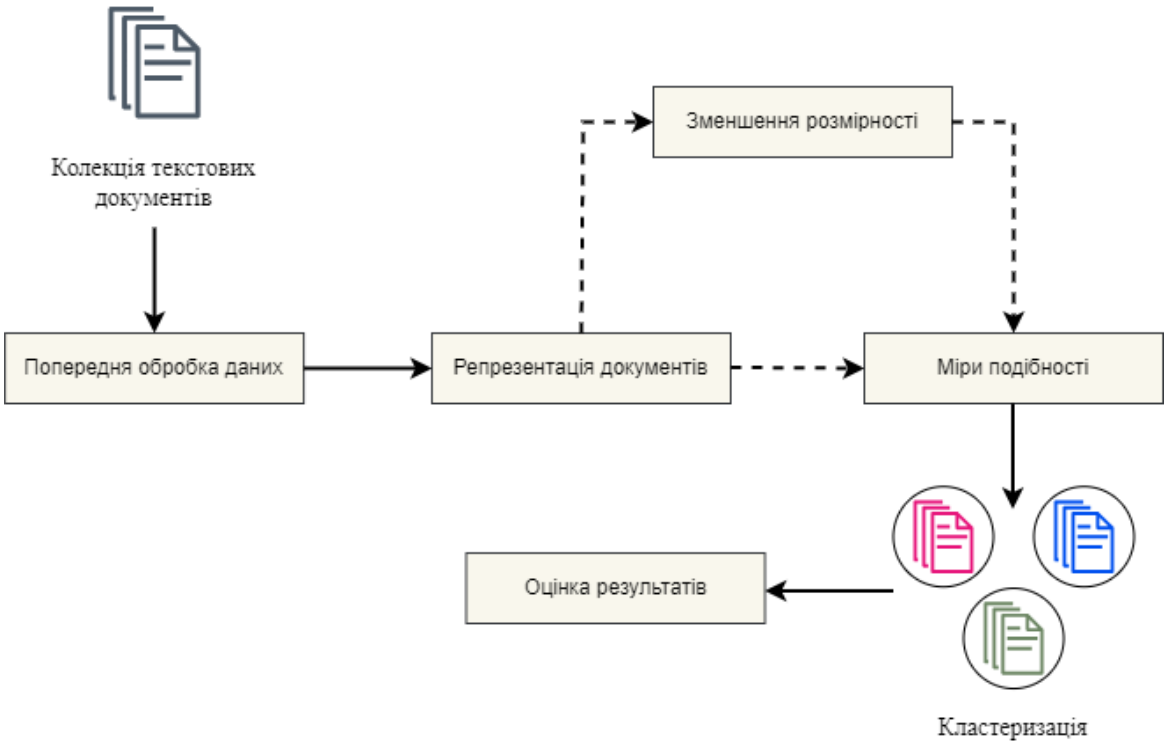

Рисунок 1.2 – Процес кластеризації текстових документів

Розглянемо детальніше кожен з компонентів діаграми зображеної на рисунку 1.2:

– попередня обробка. Це перший крок у роботі з текстовою кластеризацією. Дані повинні бути очищені від непотрібних символів, слів, знаків та цифр. Після цього можуть бути застосовані методи репрезентації тексту. Попередня обробка відіграє важливу роль у побудові ефективної системи кластеризації, оскільки вихідний текст, зазвичай, може не підходити для безпосереднього використання у кластеризації;

– репрезентація. Документи та тексти – це колекції неструктурованих даних. Ці неструктуровані дані потрібно перетворити в структурований простір ознак для застосування математичного моделювання під час кластеризації. Стандартні техніки представлення тексту можна розділити на методи представлення на основі корпусу та методи представлення на основі зовнішніх знань;

– зменшення розмірності. Тексти або документи, після репрезентації за допомогою традиційних технік, в більшості випадків стають високорозмірними. Процеси кластеризації даних можуть бути уповільнені через значний час обробки та складність зберігання. Зменшення розмірності є стандартним методом вирішення цього виду проблем. Багато хто використовує зменшення розмірності, щоб зменшити час виконання своїх програм та складність зберігання, не ризикуючи падінням продуктивності;

– міра подібності. Це основний елемент в алгоритмах кластеризації. Вона спрощує ідентифікацію подібних сутностей, групування елементів, які найбільш схожі, та визначення найкоротшої відстані між пов’язаними сутностями. Іншими словами, відстань та подібність мають обернену залежність, тому вони використовуються взаємозамінно. Векторне представлення елементів даних зазвичай використовується для обчислення мір подібності/відстані;

– техніки кластеризації. Ключовою частиною будь-якої системи кластеризації тексту є вибір найкращого алгоритму. Ми не можемо вибрати найкращу модель для системи кластеризації тексту без глибокого концептуального розуміння кожного підходу. Метою алгоритмів кластеризації є генерація внутрішньо зв’язаних кластерів, які вочевидь відрізняються один від одного;

– оцінка результатів. Це останній крок алгоритму. Необхідно оцінити отримані результати за допомогою метрик оцінки кластеризації текстових документів. Кластери можна оцінити за допомогою таких метрик, як чистота, однорідність, F-міра, V-міра та скоригований індекс Ренда (ARI) [11–13].

1.2.2 Особливості попередньої обробки текстових даних

Попередня обробка тексту є дуже важливим етапом у техніках інтелектуального аналізу тексту. Текстові документи, імовірно, містять усі типи рядків, наприклад цифри, символи, слова та фрази. Зашумлені рядки можуть негативно вплинути на продуктивність кластеризації та на пошук інформації [14]. Передня обробка включає в себе чотири основні кроки: токенізація, нормалізація, вилучення стоп-слів та стемінг. Ці кроки показані на рисунку 1.3.

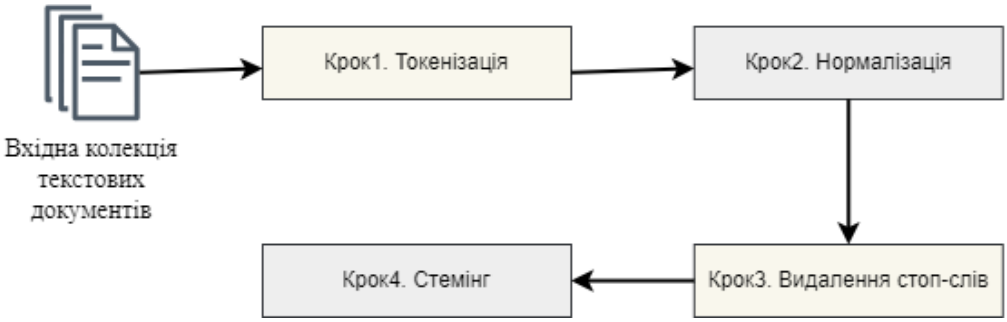

Рисунок 1.3 – Попередня обробка текстових документів

Токенізація – це процес розділення тексту на окремі значущі частини, які називаються токенами. Інакше кажучи, вона перетворює текст з документа в дані, які можуть бути проаналізовані методами машинного навчання. Токени можуть бути словами, фразами, символами або будь-якими іншими значущими частинами тексту. Зазвичай ці алгоритми розділяють текст на окремі одиниці, додаючи пробіл або який-небудь інший характерний маркер, таким чином, щоб кожному токену могло бути призначено деяке особливе слово в тексті [13–16, 18].

Нормалізація текстових даних – це процес стандартизації та очищення тексту, видалення зайвої та шумної інформації, шляхом виконання різних операцій, таких як перетворення регістру, видалення непотрібних символів або приведення слів до їх базової форми. Основна мета нормалізації – зменшити різноманітність форм представлення тексту та підготувати його для

подальшої обробки чи аналізу. Наприклад, це може включати перетворення всіх символів у нижній регістр, видалення пунктуації, чисел або спеціальних символів для приведення слів до їх базової форми. Нормалізація допомагає знизити розмірність тексту та забезпечити однакові умови для подальшої обробки даних [16-19].

На рисунку 1.4 зображено основні кроки токенізації та нормалізації текстових даних.

Рисунок 1.4 – Кроки токенізації та нормалізації текстових даних

Метою видалення стоп-слів є прибрати зайві слова, що сприяє зменшенню обсягу тексту, який здається перенасиченим: слова загального вжитку у текстових документах, такі як артиклі, прийменники та особові займенники, які не несуть значення для змісту документа [18]. Наприклад, слова «which», «and», «you», «until» мають бути вилучені з документа, оскільки вони не є ключовими словами у контексті видобутку текстових даних. Вилучення цих часто вживаних стоп-слів з текстових документів зменшує кількість слів, які потрібно порівнювати з кожним пошуковим терміном, що значно збільшує час очікування результатів запитів без впливу на точність [13].

Стемінг – це обчислювальний процес, який стандартизує слова, скорочуючи їх до спільного кореня або «стему», видаляючи префікси та суфікси, що вказують на походження або відмінювання. Наприклад, слова «chocolates», «chocolatey» або «choco» будуть спрощені до кореневого слова «chocolate», тоді як слова «consultant», «consulting» та «consultation» будуть зменшені до стему «consult». Метою стемінгу є оптимізація різних форм слова до центрального кореня або стему, щоб спростити обробку слів, які мають схожий основний зміст. Цей процес особливо корисний у пошуку інформації, де групування слів з однаковим коренем або стемом підвищує точність відповідності документів запиту [21]. Більш того, стеми значно покращують процес кластеризації та її результати, роблячи її швидше й точніше, за рахунок зменшення розміру словника та кількості залежностей від певних форм словника [20].

1.2.3 Особливості побудови моделей семантичної репрезентації текстових даних

Семантична репрезентація текстів – це спосіб представлення семантичного змісту текстової інформації у вигляді числових або векторних форм. Ця фаза є важливою, оскільки вона зводить міст між обробленим текстом та здатністю машини інтерпретувати, аналізувати та витягувати значення з цього тексту. Основна його мета полягає в тому, щоб комп’ютер зміг розуміти семантичні аспекти тексту, такі як значення слів, фраз, речень чи навіть документів. Для досягнення цієї мети використовуються різні методи та моделі машинного навчання.

У тексті та текстових документах існують два підходи до репрезентації. Один базується на синтаксичних характеристиках, він ґрунтується на лінгвістичній теорії, граматиці та логіці. Іншими словами, це векторні моделі слів, такі як Word2Vec, GloVe, FastText тощо. Ці моделі навчаються

перетворювати слова у вектори у такий спосіб, щоб семантично схожі слова мали близькі векторні представлення [22, 23].

Інший – семантичний підхід, ґрунтується на статистичних техніках та навчанні. Такою моделлю для представлення семантики речень та текстів у векторному просторі є модель BERT (Bidirectional Encoder Representations from Transformers). Вона може створити семантичне представлення для всього тексту або його фрагментів [23].

Ці два підходи відрізняються в тому, як вони можуть включати і приховувати більше або менше різних атрибутів варіації за даними.

1.2.4 Міри семантичної подібності

Семантична схожість базується на мірі, яка використовується для обчислення ступеня схожості між двома концепціями на основі ймовірності їх значень. Концепціями можуть бути речення, слова або абзаци. Вона визначає відстань між різними концепціями у семантичному просторі таким чином, що чим менше відстань, тим більша схожість. Іншими словами, семантична схожість виявляє спільні характеристики, або так звані смислові зв’язки між об’єктами, навіть якщо їх лексичні представлення різняться. [24, 25].

Більш сучасні методи обчислення семантичної схожості дозволяють враховувати не лише прямі зв’язки між словами чи фразами, а й приховані смислові залежності, які часто не очевидні для традиційних алгоритмів. Наприклад, такі моделі, як BERT, здатні розуміти нюанси значень слів залежно від їхнього контексту, що допомагає значно підвищити точність аналізу текстів.

Існуючі методи вимірювання семантичної подібності текстів показано на рисунку 1.5. Методи можна розділити на чотири основні групи: ті, що ґрунтуються на рядках, ті, що ґрунтуються на корпусі, ті, що ґрунтуються на знаннях, а також гібридні методи. Розглянем кожен з них окремо.

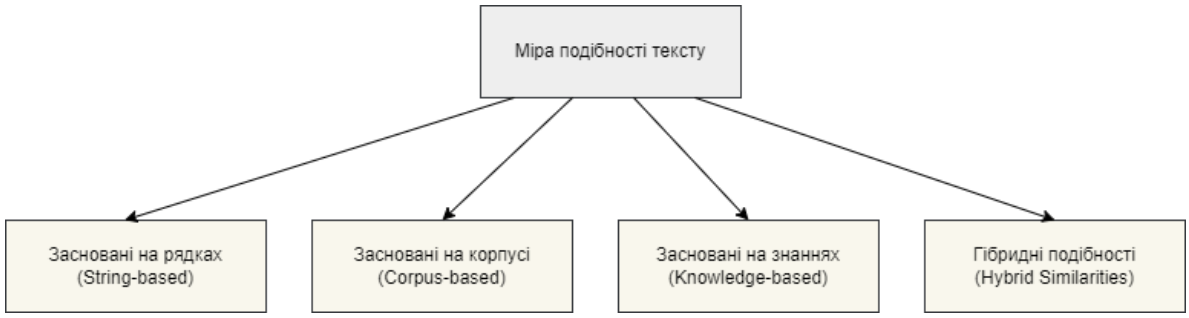

Рисунок 1.5 – Методи вимірювання подібності текстів

Подібність на основі рядків є найстарішим, найпростішим, та досі найпопулярнішим підходом до визначення подібностей. Ця міра працює з послідовностями рядків і композицією символів. Два основних типи функцій визначення подібності на основі рядків – це функції на основі символів і функції на основі токенів [26].

Схожість на основі символів також називається ще послідовнісною або вимірюванням редагування. Вона бере дві рядки символів і потім обчислює відстань редагування (включаючи вставку, видалення і заміну) між ними. Схожість на основі символів кількісно характеризує схожість символів між двома рядками для кількісної характеристики подібності, наприклад відстань редагування є мінімальною кількістю операцій редагування одного символу, необхідних для перетворення одного рядка у інший. Два рядки подібні, якщо мінімальна кількість операцій редагування менше заданого порога [27, 28]. Деякі приклади цього підходу включають відстань Хеммінга, відстань Левенштейна, Дамерау-Левенштейна, найдовшу спільну підпослідовність та N-грами. Ці підходи використовують, коли семантична інформація менш важлива, а точність та порядок символів мають вирішальне значення.

Основна характеристика схожості на основі токенів полягає в використанні перекриття двох множин токенів як кількісного показника схожості. Перекриття обчислюється на основі пар токенів, які точно співпадають, без врахування інших схожих токенів [27]. До прикладів цих методів можна віднести схожість Жаккара, коефіцієнт Дайса, косинусну схожість, манхеттенську відстань і евклідову відстань.

Загальна особливість методів вимірювання подібності текстів на основі корпусу полягає в тому, що вони розглядають текст як набір слів або термінів, і використовують статистичні методи для визначення схожості між ними. Взагалі, термін «корпус» відноситься до колекції текстових даних, які використовуються для навчання моделі або для визначення статистичних характеристик мови. Корпус може включати в себе текстові документи, новини, книги, веб-сторінки, соціальні медіа повідомлення та інші джерела текстів. Методи вимірювання подібності текстів на основі корпусу використовують зовнішній корпус для вилучення взаємозв'язку між словами або текстом. Інакше кажучи, обчислення подібності на основі корпусу передбачає перевірку подібності тексту шляхом порівняння уявлень із збіркою тексту. Деякі методи отримують взаємозв'язок між словами з великого корпусу, а потім агрегують цей взаємозв'язок для вимірювання подібності на більш високому рівні або рівні речення. Інші методи можуть безпосередньо вимірювати подібність тексту без процесу масштабування [29].

Методи вимірювання семантичної подібності, які використовують інформацію з семантичних мереж для визначення ступеня подібності слів, називаються мірами подібності на основі знань. Міри подібності на основі знань складаються з семантичної подібності та семантичного відношення. Такі міри можуть використовувати різні підходи, такі як аналіз структури знань, ієрархічні зв'язки, частоту вживання термінів тощо. Методи на основі знань використовують базу знань зі стандартизованою системою організації для обчислення подібності тексту, як правило, розділену на знання на основі онтології та знання на основі мереж. Зазвичай побудова онтології вимагає участі експертів, що забирає багато часу. При використанні онтології для обчислення подібності тексту спочатку обчислюють на рівні слів, а потім накопичують подібності слів, щоб отримати подібність довгого тексту. Порівняно з методами, що базуються на корпусах, ефективність обчислення довгих текстів низька. Методи ж на основі мережі, прикладом яких є алгоритми, що використовують мережеве знання з платформ, таких як

Вікіпедія, мають переваги у структурованій інформації, покращуючи розуміння комп’ютера і дозволяючи точні обчислення подібності тексту. Однак вони залежать від зовнішніх джерел, таких як Вікіпедія, яка може не однаково охоплювати всі галузі або містити упереджену інформацію [29–34].

Окрім трьох вище описаних категорій, існує кілька мір подібності, які не можуть бути віднесені до жодної з раніше визначених груп. Ідея цього підходу полягає в поєднанні раніше описаних методів, включаючи методи на основі рядків, корпусів та знань, щоб отримати кращу метрику, використовуючи їх переваги. Загальні приклади гібридних метрик включають метод Level2, запропонований Монжем і Елканом, SoftTFIDF, узагальнену метрику редагування (GES) та підходи Ванга та інші.

1.3 Використання семантичної кластеризації в сфері новин

Використання семантичної кластеризації в сфері новин полягає у застосуванні методів аналізу текстів для групування новинних статей за їхнім змістом та семантикою. Це дозволяє автоматично організувати великий обсяг новинних матеріалів у логічні та зрозумілі групи, що сприяє зручнішому доступу до інформації для користувачів. Тобто, результатом проведення семантичної кластеризації новини розподіляються у відповідні кластери на основі їхньої семантики, що допомагає швидше знаходити новини за відповідною тематикою та зрозуміти загальний зміст новинного потоку.

1.3.1 Проблематика врахування семантичних зв’язків у новинах

Під час роботи з новинами можна зустрітись із рядом різних аспектів, які можуть впливати на точність та повноту розуміння викладеної інформації. Отже, для досягнення вищих результатів необхідно враховувати особливості

семантики, адже вони можуть впливати на його інтерпретацію. Ось кілька можливих проблем, які можуть виникнути при роботі з новинами [35]:

– полісемія та омонімія. Полісемія відноситься до явища, коли одне слово має декілька різних значень, які можуть бути контекстуально залежними. Наприклад, англійське слово «business» має такі значення: бізнес, фірма, торгівельне підприємство, справа, професія, обов'язок, право. Омонімія, з іншого боку, це явище, коли два різні слова мають однакове звучання, але різне значення. Наприклад, слово кран, як механізм для підіймання й переміщення вантажів і кран, як трубка з закривкою, якою перекриваємо воду, газ тощо. У новинах, де точність та чіткість мови є важливими, розуміння контексту та розрізнення між різними значеннями слів є критичним;

– саркастичний та іронічний контекст. Саркастичні або іронічні висловлювання можуть бути складними для автоматичного розпізнавання, оскільки вони можуть протистояти прямій інтерпретації тексту. Такі висловлювання можуть впливати на загальний сенс новин та вимагати спеціалізованих алгоритмів для їх розпізнавання;

– семантична та лексична неоднозначність. Багатозначність слів або фраз може призводити до неоднозначності семантичного змісту тексту;

– динаміка мовлення. Мова новин швидко змінюється, включаючи появу нових термінів, сленгу або технічної лексики, що ускладнює підтримку актуальності моделей аналізу;

– контекстуальна залежність. Значення слова чи виразу може змінюватися в залежності від контексту, у якому воно вживається. Наприклад, вживання слова «замок» в різних контекстах може вказувати на будівлю або на механізм для запирання чого-небудь.

Таким чином, для досягнення високої точності та повноти розуміння новинного контенту важливо поєднувати технологічний підхід з експертними знаннями, що дозволить ефективно враховувати різноманітні аспекти семантики у процесі аналізу новин.

1.3.2 Перспективи розвитку семантичної кластеризації новин

Майбутнє семантичної кластеризації новин обіцяє багато перспективних напрямків розвитку, які можуть суттєво змінити спосіб, яким ми сприймаємо, аналізуємо та використовуємо новинний контент.

Перше, що спадає на думку при обмірковуванні перспектив розвитку кластеризації новин – її використання з метою персоналізації новин. Це є достатньо перспективним напрямком розвитку семантичної кластеризації, що може значно покращити спосіб, яким користувачі отримують та споживають новини. Основна ідея полягає в тому, щоб система автоматично адаптувалася до індивідуальних потреб та вподобань кожного користувача, надаючи змогу персоналізувати новинний досвід для кожного.

Ще одною, не менш важливою сферою використання семантичної кластеризації новин є боротьба з дезінформацією та фейковими новинами. Розвиток семантичних алгоритмів аналізу може стати потужним інструментом у цій боротьбі, надаючи засоби для ефективного виявлення та фільтрації недостовірної інформації. Семантичні алгоритми можуть допомагати відслідковувати поширення фейкових новин у мережі та ідентифікувати певні шаблони або кластери, які можуть свідчити про спроби масової дезінформації.

Також, семантична кластеризація може бути використана для підвищення ефективності розумних систем, таких як віртуальні асистенти та інші системи штучного інтелекту, що вимагають розуміння контексту та змісту новин.

1.4 Постановка задачі дослідження

Таким чином, провівши аналіз предметної області, можна наголосити про актуальність проблеми семантичної кластеризації текстових даних, а зокрема новин. Актуальність роботи полягає у важливості фільтрування

великого потоку інформації, а також існуванні ряду проблем, пов’язаних з роботою з новинами та необхідністю пошуку оптимальних підходів роботи з ними для автоматизованого аналізу новинних потоків, підвищення ефективності та точності процесу їх групування за змістом.

Об’єктом дослідження є методи семантичної кластеризації, що можуть бути використані в роботі з новинами.

Метою дослідження є вивчення та порівняння методів семантичної кластеризації на прикладі задачі кластеризації новин відповідно до їх тематики.

Для досягнення описаної мети необхідно вирішити наступний ряд завдань:

– провести дослідження існуючих методів семантичної кластеризації текстових даних, виділити їхні переваги та недоліки;

– дослідити роботу кожного з цих методів при однакових вхідних даних на прикладі реальних або симульованих даних;

– надати оцінку отриманих результатів та порівняти ці методи;

– візуалізувати отримані дані за допомогою графіків, діаграм, тощо;

– сформулювати висновки щодо придатності та ефективності існуючих методів семантичної кластеризації для задачі кластеризації новин на основі отриманих результатів;

– визначити перспективи подальшої роботи.

2 МАТЕМАТИЧНІ МОДЕЛІ СЕМАНТИЧНОЇ КЛАСТЕРИЗАЦІЇ ТЕКСТОВИХ ДАНИХ

Математичні моделі семантичної кластеризації текстів виступають основою для розробки ефективних алгоритмів та програмних засобів, які допомагають організовувати та аналізувати великі обсяги текстової інформації. Вони забезпечують можливість автоматизованого групування текстів за їх семантичним змістом, що дозволяє користувачам ефективно витягати корисну інформацію та здійснювати аналіз семантичних відношень між текстовими документами.

У цьому розділі роботи досліджуватимуться існуючі методи та алгоритми, що лежать в основі семантичної кластеризації текстових даних, а також виділятимуться їх переваги та недоліки в контексті кластерного аналізу.

2.1 Базовий алгоритм K-Means

K-Means – найбільш популярний алгоритм, що використовується для групування даних, розділяючи їх на K різних кластерів в залежності від подібності даних. Алгоритм був розроблений математиком Гуго Штейнгаузом у 1950-х роках та майже одночасно – Стюартом Ллойдом. Найбільшої популярності він набув після роботи Маккуіна.

Під час роботи алгоритм оновлює результати групування за допомогою ітераційного обчислення відстані між точками та центрами кластерів.

Кластеризація за K-Means полягає у групуванні всіх даних у відокремлені кластери, враховуючи схожість кожної вибірки з центрами кластерів. Випадкові вибірки визначаються як основні центри кластерів, а потім решта вибірок розподіляються в кожен кластер, який найбільш відповідає його центру. Після кластеризації кожної вибірки дані центри

кластерів оновлюються до середнього значення всіх елементів у цьому кластері. Цей процес повторюється до тих пір, поки центри кластерів не перестануть змінюватись [36–38]. Тобто, робота алгоритму складається з наступних п’яти кроків:

Крок 1. Початкова ідентифікація K кластерів: $k = 1, 2, \ldots, K$.

Крок 2. Вибір початкових центрів (центроїдів) для кожного з кластерів $c_i = (c_1, c_2, \ldots, c_k)$, де c_i – центр (центроїд) кластеру i, а x_j – елемент вибірки j:

$$c_i = x_j \text{ , для деякого } j. \tag{2.1}$$

Крок 3. Кожен з елементів вибірки x_j призначається до найближчого центру кластера згідно з функцією відстані d, наприклад за вирахуваною відстанню Евкліда:

$$min_i d(x_j, c_i). \tag{2.2}$$

Крок 4. Проведення обчислення та оновлення нових центрів кластерів. Позначимо N_i як кількість елементів кластеру i, а x_{new} – як елемент, який щойно потрапив до цього кластеру:

$$c_i = \frac{1}{1+N_i}\left(\sum_{j \in Cluster\ i} x_j + x_{new}\right). \tag{2.3}$$

Крок 5. Якщо деякі дані стають непотрібними або застарілими, може бути потрібна модифікація кількості кластерів. Кроки 2-4 повторюються до тих пір, поки центри кластерів не залишатимуться незмінними.

K-Means можна успішно використовувати в семантичній кластеризації текстових даних. Він добре підходить для групування текстів за їх семантичним змістом, особливо коли в текстах представлені уніфіковані ознаки, такі як частоти використання слова або векторні представлення

текстів. При використанні K-Means у семантичній кластеризації текстових даних важливо правильно вибрати вхідні ознаки або представлення текстів, а також підібрати оптимальну кількість кластерів.

2.1.1 Модифікації алгоритму: Bisecting K-Means та Fuzzy C-Means

Bisecting K-Means та Fuzzy C-Means – це дві варіації алгоритму K-Means, які розширюють його базовий принцип для вирішення конкретних випадків кластеризації.

Bisecting K-Means використовується для поділу кластерів на дві частини на кожному кроці ітерації замість розділення всіх даних на кожному кроці, як це робить оригінальний алгоритм K-Means (рис.2.1). Спочатку весь набір даних розглядається як один кластер, а потім він рекурсивно поділяється на два підкластери на кожному кроці, обираючи для розділення кластер, який має найбільшу сумарну внутрішню варіативність [39-41].

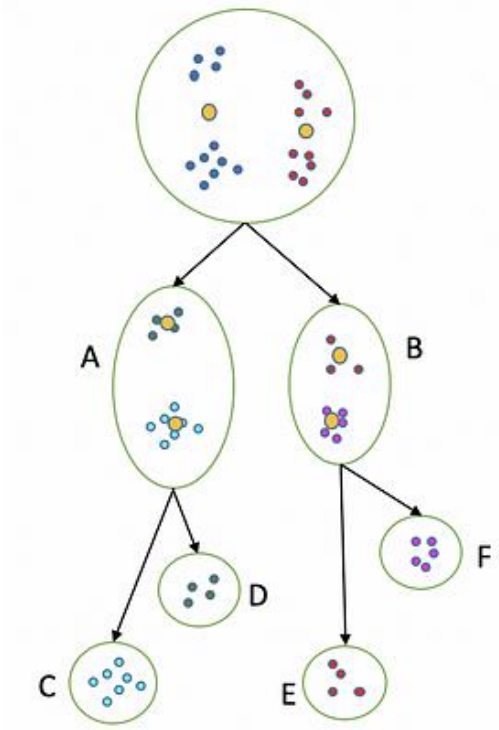

Рисунок 2.1 – Візуалізація алгоритму Bisecting K-Means

Розглянемо загальний алгоритм роботи Bisecting K-Means (рис. 2.1):

Крок 1. Встановлення K кластерів.

Крок 2. Призначення всіх вхідних даних до одного спільного кластеру.

Крок 3. Використання методу K-Means для $K = 2$, для розділення загального кластеру на два менші (кластери A та B на рисунку 2.1).

Крок 4. Вимірювання відстані для кожного внутрішнього кластера, наприклад за формулою суми квадратів:

$$\sum_{i=0}^{n}(x_i - \bar{x})^2, \tag{2.4}$$

де n – кількість елементів у кластері;

x_i – значення конкретного елемента (координата даної точки);

$\bar{x}$ – середнє значення всіх елементів кластеру, визначається як $\bar{x} = \frac{1}{n}\sum_{i=0}^{n} x_i$.

Крок 5. Вибір кластеру з найбільшим показником відстані та його поділ на два кластери, знову ж таки за допомогою методу K-Means (кластери C та D на рисунку 2.1).

Крок 6. Кроки 3–5 ітеративно повторюються, доки кількість кластерів не дорівнюватиме заданому K [41].

Таким чином, Bisecting K-Means дає кластери однакових розмірів, тоді як класичний K-Means, як відомо, створює кластери дуже різних розмірів.

Fuzzy C-Means також використовує ідею кластеризації, але замість того, щоб призначити кожен об’єкт до одного кластеру, він надає кожному об’єкту «ступінь належності» до кожного кластеру у вигляді числа від 0 до 1. Ця ступінь належності відображається у вигляді матриці, яка називається матрицею приналежності, де кожен елемент визначає ймовірність того, що даний об’єкт належить до певного кластеру [42].

Розглянемо детальніше алгоритм Fuzzy C-Means [43–47]. Для кожного елементу t існує відповідний коефіцієнт, що вказує на ступінь $u_k(t)$ приналежності до деякого k-го кластера. Зазвичай сума цих коефіцієнтів

дорівнює 1, тому $u_k(t)$ вважається ймовірністю приналежності t до певного кластера:

$$\forall t: \sum_{k=1}^{c} u_k(t) = 1. \quad (2.5)$$

У випадку використання алгоритму Fuzzy C-Means, центроїди кластерів обчислюються, як середнє значення всіх елементів, зважене за ступенем u_k приналежності до деякого кластера C_k. Це обчислення відбувається за наступною формулою:

$$center_k = \frac{\sum_t u_k^m(t) * t}{\sum_t u_k(t)}. \quad (2.6)$$

Ступінь приналежності до певного кластера пов'язана з оберненою величиною відстані до центру кластера:

$$u_k(t) \propto \frac{1}{dcenter_k(t)}, \quad (2.7)$$

де $u_k(t)$ – ступінь приналежності точки t до кластера k;

$\propto$ – символ пропорційності, який означає, що $u_k(t)$ прямо залежить від оберненої величини відстані до центру кластера;

$dcenter_k(t)$ – відстань від точки t до кластера k.

При цьому відстань може обчислюватися за різними метриками, найпоширенішою з яких є Евклідова відстань.

Самі коефіцієнти фазифікуються з реальним параметром $m > 1$, а також нормалізуються таким чином, щоб їх сума дорівнювала 1:

$$u_k(t) = \frac{1}{\sum_j \frac{dcenter_k(t)}{dcenter_j(t)}^{\frac{2}{m}-1}}. \quad (2.8)$$

Використання методу Fuzzy C-Means дозволяє отримати більш гнучкі результати кластеризації, де об’єкти можуть належати до кількох кластерів одночасно з різними ступенями вірогідності.

2.1.2 Переваги та недоліки K-Means та його модифікацій

Звичайний K-Means, як правило, простий в реалізації та ефективний для використання з наборами даних, де кластери мають прості форми та однакову кількість елементів у кожному кластері. Однак, він може виявити складності з кластеризацією даних, де кластери мають різну густину або форму, а також в присутності викидів або шуму у даних.

Bisecting K-Means може бути корисним, коли важко визначити оптимальну кількість кластерів заздалегідь або коли потрібно уникнути застрягання в локальних мінімумах. Він зазвичай потребує більше обчислювальних ресурсів порівняно зі звичайним K-Means, оскільки кожна ітерація включає в себе поділ кластера на два [39]. Bisecting K-Means ефективніший, коли значення K велике. Для алгоритму K-Means обчислення включає кожну точку даних набору даних і k центроїдів. З іншого боку, на кожному кроці Bisecting K-Means в обчисленні беруть участь лише точки даних одного кластера та двох центроїдів. Таким чином, час обчислень значно скорочується.

Fuzzy C-Means є найбільш доцільним у використанні для випадків, коли об’єкти можуть належати до кількох кластерів або коли нечітко визначена належність до кластерів більш підходить для представлення даних. Однак цей метод може бути чутливим до параметрів, таких як кількість кластерів та рівень розмитості, і його результати можуть бути важко інтерпретувати. Також, обчислення цим алгоритмом можуть займати тривалий час [43-45].

У таблиці 2.1 наведено порівняльні характеристики алгоритмів K-Means, Bisecting K-Means та Fuzzy C-Means.

Таблиця 2.1 – Порівняльні характеристики алгоритмів

Характеристика	**K-Means**	**Bisecting K-Means**	**Fuzzy C-Means**
Складність реалізації	Середня	Висока	Висока
Кількість кластерів	Попередньо задана		
Гнучкість	Низька	Середня	Висока
Вразливість шумом	Вразливий	Витримує шум	Витримує шум
Потреба в пам’яті	Низька	Висока	Висока
Час обчислень	Середній	Малий	Тривалий

Окрім того, кожен з цих методів може бути чутливим до початкових умов, і успішність кластеризації може залежати від правильного вибору початкових центрів кластерів. Неправильно обрана кількість кластерів може призвести до неправильних або неінтерпретованих результатів кластеризації.

2.2 Алгоритми ієрархічної кластеризації. Їх переваги та недоліки

Ієрархічна кластеризація – це метод аналізу даних, під час роботи якого створюється ієрархічне представлення кластерів у наборі даних. Метод починається з обробки кожної точки даних як окремого кластера, а потім ітеративно об’єднує найближчі кластери, доки не буде досягнуто критерій зупинки [46-48]. В результаті створюється дендрограма – деревоподібна структура, що ілюструє ієрархічні зв’язки між кластерами, приклад якої наведено на рисунку 2.2.

Ієрархічна кластеризація володіє декількома перевагами перед іншими методами семантичної кластеризації. Вона може працювати з неопуклими кластерами, а також з кластерами різного розміру та щільності. Крім того, цей метод може обробляти відсутні дані і дані з шумом, а також розкривати

ієрархічну структуру вхідних даних, що допомагає в розумінні зв’язків між кластерами.

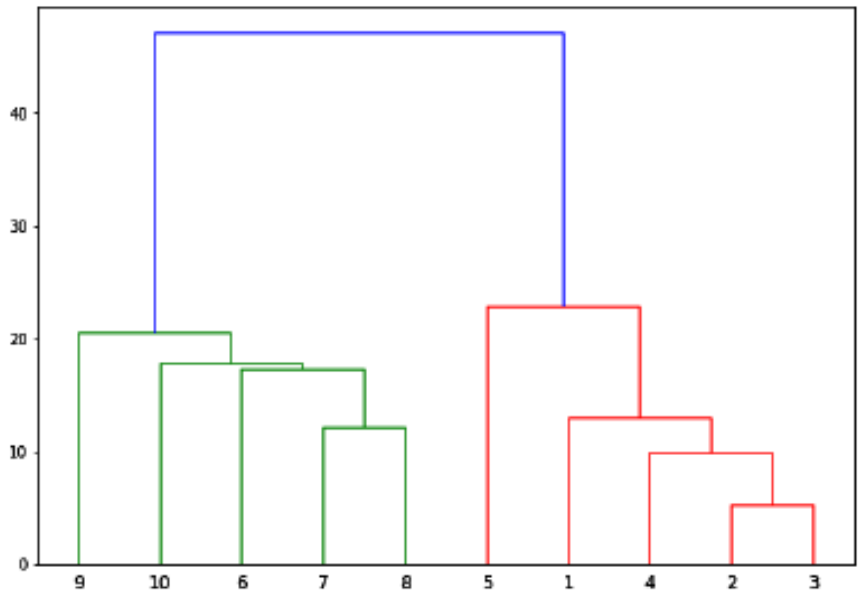

Рисунок 2.2 – Приклад дендрограми

Не дивлячись на переваги, ієрархічна кластеризація має свої недоліки. Необхідно встановлювати критерій зупинки процесу кластеризації та визначати остаточну кількість кластерів, що може бути складним завданням. Обчислювальні витрати і вимоги до пам’яті цього методу можуть бути високими, особливо для великих наборів даних. Крім того, результати можуть бути чутливими до початкових умов, критерію зв’язку та використовуваної метрики відстані [48–51].

Існує два підходи виконання ієрархічної кластеризації – агломератовий (об’єднувальний) та розділювальний. Розглянемо обидва ці методи більш детально.

HAC (Agglomerative Hierarchical Clustering) вважається методом «знизу-вгору». У цій техніці спочатку кожна точка даних розглядається як окремий кластер. На кожній ітерації подібні кластери зливаються з іншими кластерами, поки не буде сформований один або K кластерів. Алгоритм складається з наступних кроків:

Крок 1. Кожен елемент призначається одному кластеру.

Крок 2. Проводиться вимірювання відстані та обчислюється матриця відстані.

Крок 3. Визначаються критерії «зв'язку» для об'єднання кластерів.

Крок 4. Оновлюється матриця відстаней.

Крок 5. Процедура повторюється, доки не сформується однин спільний кластер [50–53].

HDC (Divisive Hierarchical Clustering) – прямо протилежний методу HAC: всі елементи розглядаються як один кластер і на кожній ітерації відокремлюються несхожі елементи, кожен з яких розглядається як окремий кластер. Зрештою, має залишитись n кластерів [51, 52].

2.3 Прихований розподіл Діріхле. Переваги та недоліки

Latent Dirichlet Allocation (LDA) – це ймовірнісний метод тематичного моделювання, який використовується для темо-орієнтованої моделі текстових даних, таких як статті, новини, блоги, тощо. LDA вважається формою неконтрольованого навчання, яка використовується для виявлення латентних (прихованих) угруповань у даних. Виявлені угруповання часто називають темами. В контексті текстової кластеризації алгоритм LDA допомагає виявляти приховані теми в колекції документів і розподіляти ці теми серед документів та слів (рис. 2.3), представляючи документи як суміші тем з певними ймовірностями.

У алгоритмі LDA використовується два розподіли Діріхле. У випадку аналізу документів, один розподіл задає теми в кожному документі, а інший представляє слова в кожній темі. При використанні LDA необхідно заздалегідь визначити кількість тем, так само як при використанні k-means потрібно вказати кількість кластерів (k). Розподіли тем і слів в LDA ґрунтуються на підрахунках частот тем і слів у наборі документів, що аналізуються [54–56].

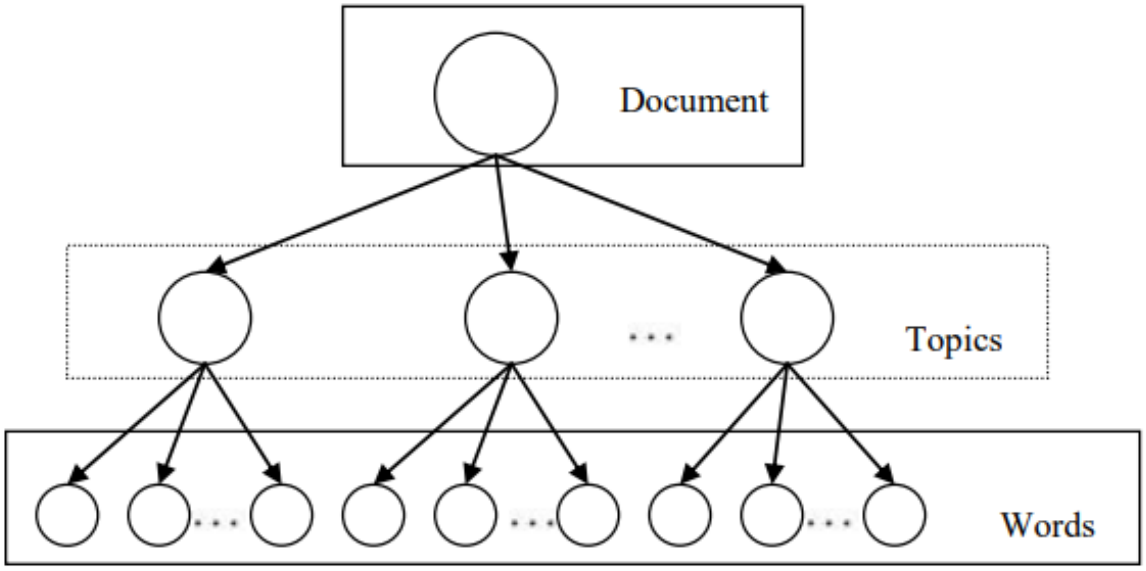

Рисунок 2.3 – Топологія мережі прихованих тем LDA

Нижче наведено формулу LDA:

$$\varphi_{i,j} = p(w_i|z_j), \tag{2.9}$$

$$\theta_{i,j} = p(m_i|z_j), \tag{2.10}$$

$$p(w, z, \theta, \varphi|\alpha, \beta) = p(\varphi|\beta)p(\theta|\alpha)p(z|\theta)p(w|\varphi z), \tag{2.11}$$

де w – слова;

z – теми;

m – документи;

α – параметр розподілу Діріхле для тем у документах;

β – параметр розподілу Діріхле для слів у темах.

З огляду на документ, основною проблемою висновку LDA є обчислення апостеріорного розподілу прихованих змінних, що є складною проблемою. Щоб впоратися з цією важкорозв'язною проблемою, використовуються такі алгоритми апроксимації, як апроксимація Лапласа, варіаційна апроксимація, вибірка Гіббса та ланцюг Маркова Монте-Карло.

Найбільш поширений метод оцінки моделі LDA є вибірка Гіббса. Імовірність того, що слово в документі пов'язане з темою j, можна виразити таким чином:

$$P(z_i = j | z_{-i}, w_i, d_i) \propto \frac{C_{w_i j}^{WT} + \eta}{\sum_{w=1}^{W} C_{wj}^{WT} + W\eta} \frac{C_{d_i j}^{DT} + \alpha}{\sum_{t=1}^{T} C_{d_i t}^{DT} + T\alpha}, \quad (2.12)$$

де W – довжина словника (тобто кількість унікальних слів);

T – кількість тем;

C_{dj}^{WT} – матриця, що містить кількість разів, скільки тема z призначена для певного слова w в документі d;

C_{wj}^{WT} – матриця, що містить кількість разів, скільки слово w призначене для теми z;

α – гіпер-параметр. Низьке значення α збільшує ймовірність того, що кожен документ буде складатися з небагатьох домінуючих тем, тоді як високе значення повертає значно більше домінуючих тем;

η – гіпер-параметр. Низьке значення η надає більше ваги темам, що складаються лише з кількох домінуючих слів [57–59].

До переваг LDA можна віднести те, що він дозволяє проводити тематичне моделювання, виявляючи приховані теми у великих текстових масивах, що сприяє кращому розумінню їх змісту та структури. LDA відзначається гнучкістю та може застосовуватися до різних видів текстових даних, зокрема наукових статей, оглядів продуктів і соціальних мереж. Він є некерованим методом і не потребує ручного маркування даних, що робить його особливо корисним для аналізу великих неструктурованих корпусів.

Цей алгоритм має ряд досить вагомих недоліків. По-перше, потрібно заздалегідь визначити кількість тем, цей вибір може бути непростим і впливати на кінцеві результати. По-друге, обробка великих корпусів може вимагати значних ресурсів, особливо під час виконання вибірки за методом Гіббса. По-третє, LDA досить чутливий до гіпер-параметрів, таких як α і η, а

правильне налаштування цих параметрів потребує досвіду та експериментів. І, наостанок, результати LDA можуть бути складними для інтерпретації, особливо якщо теми мають складну структуру або перетинаються.

2.4 Особливості LSI/LSA

Latent Semantic Analysis (LSA), також відомий як Latent Semantic Indexing (LSI), є методом обробки природної мови та аналізу текстів, який використовується для виявлення прихованих відносин і шаблонів у великих наборах текстових даних. LSI/LSA обчислює ортогональні виміри матриці термінів-документів, і за допомогою сингулярного розкладу (SVD), про який йтиметься далі, LSA отримує проекцію тексту в низьковимірному підпросторі, на основі чого можна обчислити схожість між термінами чи документами шляхом підрахунку кутів між векторами у просторі після проекції, а також виявляти зв'язок між ними.

Таким чином, будь-яка прямокутна матриця термінів-документів розміром $M \times N$, позначена як C з рангом r, може бути розкладена за допомогою сингулярного розкладу як:

$$C = USV^T, \tag{2.13}$$

де $S = diag(\lambda_1, \dots, \lambda_r)$ та $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r$ – сингулярні значення матриці;

$U = [u_1, \dots, u_r]$ та u_i – лівий сингулярнй вектор;

$V = [v_1, \dots, v_r]$ та v_i – правий сингулярнй вектор.

Практичне значення SVD виглядає наступним чином. U_k та V_k – матриці з k-вимірного простору, отримані з початкової матриці термінів і документів C, що використовуються як координати точок для термінів і документів відповідно. Оскільки k – кількість обраних тем, кожен рядок U_k розглядається як термін, де кожен вимір відображає ступінь кореляції між терміном та

темами. Чим вище значення у вимірі, тим сильніша кореляція між терміном і відповідною темою. Аналогічно, кожен рядок V_k можна розглядати як документ, де кожен вимір представляє ступінь кореляції між документом і темами. Чим вище значення у вимірі, тим важливіший документ для тієї чи іншої теми. Крім того, S_k можна розглядати як показник ступеня кореляції між термінами та документами: чим вище значення, тим сильніша кореляція термінів і документів для цієї теми.

Таким чином, всі терміни або документи можна класифікувати на k кластерів, де k – обрана кількість тем.

Отже, алгоритм виконання кластеризації за допомогою LSI/LSA виглядає наступним чином:

Крок 1. Завдання матриці «термін-документ» C, розмірністю $M \times N$.

Крок 2. Здійснення SVD над матрицею.

Крок 3. Встановлення значення k та знаходження мінімуму, що задовольнятиме нерівність:

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{j=1}^{rank(C)} \lambda_j} \geq \delta, \tag{2.14}$$

де δ – порогове значення (міра, що визначає відносний внесок вибраних сингулярних значень).

Крок 4. Здійснити k-SVD для отримання k-вимірних координат для тегів та ресурсів за формулою $C = U_k S_k {V_k}^T$.

Крок 5. Призначити терміни та документи кластерам, використовуючи відповідно U_k та V_k [38].

До очевидних переваг LSI/LSA відносять незалежність від знань про мову. SVD виконує лише математичну обробку матриці, а значить не потребує жодних знань граматики, семантики або інших понять NLP, що робить метод досить гнучким і простим. Також, за допомогою цього методу вирішується проблема рідкісності даних: у початковій матриці слово може з’являтися лише

в кількох документах (це призводить до великої кількості нульових значень, ускладнюючи роботу з матрицею), а після SVD вимірність простору значно зменшується, що частково вирішує проблему рідкісності даних.

Попри те, що LSI ефективно зменшує розмірність даних, проблема синонімів не вирішується, оскільки процес зменшення розмірності негнучкий і не враховує попередню інформацію. Більш того, процес SVD має високу обчислювальну складність, що ускладнює використання алгоритму у масштабних даних та реальних застосуваннях із сучасними обчислювальними потужностями.

2.5 Спектральна кластеризація

Спектральна кластеризація – це варіант алгоритму кластеризації, що використовує зв’язок між точками даних, для формування кластерів. Використовуються власні значення та власні вектори матриці даних для передбачення даних у простір меншої розмірності. Цей метод базується на ідеї графічного представлення даних, де точки даних відображаються як вузли, а подібність між цими точками відображається ребрами.

Спектральна кластеризація базується на 3 основних етапах, а саме: побудова графу подібності, проектування даних на простір з меншою розмірністю та кластеризація даних.

На етапі побудови графу подібності даних створюється граф подібності у вигляді деякої матриці суміжності A, що будується одним із наведених нижче способів.

Першим способом побудови графу являється побудова графу епсілон-околу (epsilon-neighbourhood graph). В цьому випадку спочатку фіксується деяке значення параметру епсилон, і кожна точка даних з’єднується з усіма точками, що знаходяться в її так званому «епсилон-радіусі». Якщо всі відстані між будь-якими двома точками подібні за масштабом, то зазвичай ваги ребер

(тобто відстань між двома точками), не зберігаються, оскільки вони не надають ніякої додаткової інформації. Таким чином, у цьому випадку побудований граф є неорієнтованим і незваженим.

Другий варіант – побудова графу за допомоги K-найближчих сусідів. Тут, заздалегідь вказується параметр k, після чого для двох вершин u і v, ребро вважається орієнтованим від u до v лише тоді, коли v знаходиться серед K-найближчих сусідів u. Отже, формується зважений і орієнтований граф, бо не для кожного u, у якого v є одним з K-найближчих сусідів, знайдеться такий v, у якого u буде одним з K-найближчих сусідів. В цьому випадку перетворити граф на неорієнтований можна одним з наступних підходів:

– спрямувати ребро від u до v і від v до u, якщо v є K-найближчим сусідом u АБО u є K-найближчим сусідом v;

– спрямувати ребро від u до v і від v до u, якщо v є K-найближчим сусідом u І u є K-найближчим сусідом v.

Третій, і останній, варіант побудови графу подібності – побудова повнозв'язного графу, для цього кожна точка з'єднується з неорієнтованим ребром, реберно-зваженого відстанню між двома точками, до кожної іншої точки. Зазвичай для обчислення відстані використовується материка схожості за гаусом, оскільки такий підхід використовується для моделювання локальних «сусідських» відносин.

Процес проектування даних на простір з меншою розмірністю полягає в урахуванні можливого віддаленого розміщення елементів кластера один від одного в розглядуваному просторі. Розмірність цього простору зменшується таким чином, щоб точки з одного кластеру знаходились ближче між собою, що дозволить кластеризувати їх разом за допомогою традиційних методів. Такий розрахунок відбувається шляхом обчислення графової матриці Лапласа [60].

До переваг спектральної кластеризації слід віднести масштабованість: оскільки перед безпосередньою кластеризацією відбувається зменшення розмірності, алгоритм дозволяє обробляти достатньо великий обсяг даних. Також слід відзначити, що алгоритм є гнучким та є придатним для

застосування до нелінійно-роздільних даних, оскільки не ґрунтується на традиційних методах кластеризації, заснованих на відстанях. Більш того, спектральна кластеризація є більш стійкою до шумів і викидів у вхідних даних, оскільки розглядається глобальна структура, а не лише локальні відстані між точками даних.

Спектральна кластеризація, попри всі свої переваги, має також недоліки. Один з головних недоліків полягає в обчислювальній складності, особливо у разі великих наборів даних через необхідність обчислення власних векторів та власних значень. Крім того, вибір числа кластерів та належної матриці схожості є складною задачею, що може потребувати додаткової експертизи, або здійснення експериментів задля досягнення бажаних та оптимальних результатів.

2.6 Алгоритм DBSCAN. Його особливості, переваги та недоліки

DBSCAN – це перший алгоритм кластеризації даних, що базується на щільності. Він був запропонований Естером в 1996 році і призначений для кластеризації даних будь-яких форм у випадку наявності шуму в просторових і не просторових високо-вимірних даних. Він дозволяє виявляти кластери будь-якої форми та розміру, а також ідентифікувати так звані шумові точки.

Перш за все, DBSCAN визначає два параметри: радіус кластера ε та значення $MinPts$. Перший представляє щільність кластера, яка виражена середньою відстанню між кластерами. Другий – мінімальну кількість об’єктів в межах одного кластера. Базові концепції алгоритму виглядають так:

– густота будь-якої точки в просторі є кількістю точок в круглій області, де центр кола – деяка точка, а його радіус – ε;

– ε-сусідство будь-якої точки в просторі – точки в круглій області, де де центр кола – деяка точка, а його радіус – ε. Сусідство ε-околу довільної точки p визначається як:

$$N_{\varepsilon} = \{q \in \frac{D}{dist(p,q)} < \varepsilon\}, \quad (2.15)$$

де D – набір вхідних даних;

– ядро – будь-яка точка в просторі, густота якої більша за заданий поріг $MinPts$:

$$N_{\varepsilon}(P) > MinPts\}; \quad (2.16)$$

– граничний об’єкт – будь-яка точка в просторі, густота якої менша за заданий поріг $MinPts$;

– поняття прямої щільнісної досяжності до об’єкту. Нехай, задано деякий набір даних D, а також ε та $MinPts$, і p, q, що належать D. Якщо q – ядро, а p – ε-сусід q, то p є прямо щільнісно досяжний від q;

– поняття щільнісної досяжності. Нехай, задано деякий набір даних D, ε та $MinPts$. Якщо існує деякий ланцюг $p_i \in D(1 \leq i \leq n)$, і $p_1 = q$, $p_n = p$, і p_i досягає p_{i+1} напряму, то p є щільно досяжний від q;

– поняття щільнісної спорідненості. Нехай, задано деякий набір даних D, а також ε та $MinPts$, і p, q, що належать D. Якщо існує об’єкт r, а p та q є щільнісно досяжними від r, то p та q вважаються щільнісно спорідненими;

– кластери представлені колекцією максимально щільнісно споріднених точок на основі прямої щільнісної досяжності;

– шум – об’єкти, які не включені до жодного кластера [55].

На рисунку 2.4 можна наглядно побачити деякі з описаних вище понять.

Зважаючи на перелічені вище поняття, алгоритм роботи DBSCAN можна описати наступним чином. DBSCAN шукає кластери, перевіряючи ε-околи кожного об’єкта набору даних. Якщо ε-окіл об’єкта p містить більше точок, ніж заданий $MinPts$, створюється новий кластер, де p являється ядром. Потім ітеративно накопичуються об’єкти, які прямо щільнісно досягають цих

ядер, що може включати утворення нового кластера, досяжного за щільністю. Кінцевою умовою цього алгоритму є те, що до жодного з утворених кластерів не можна додати нові точки.

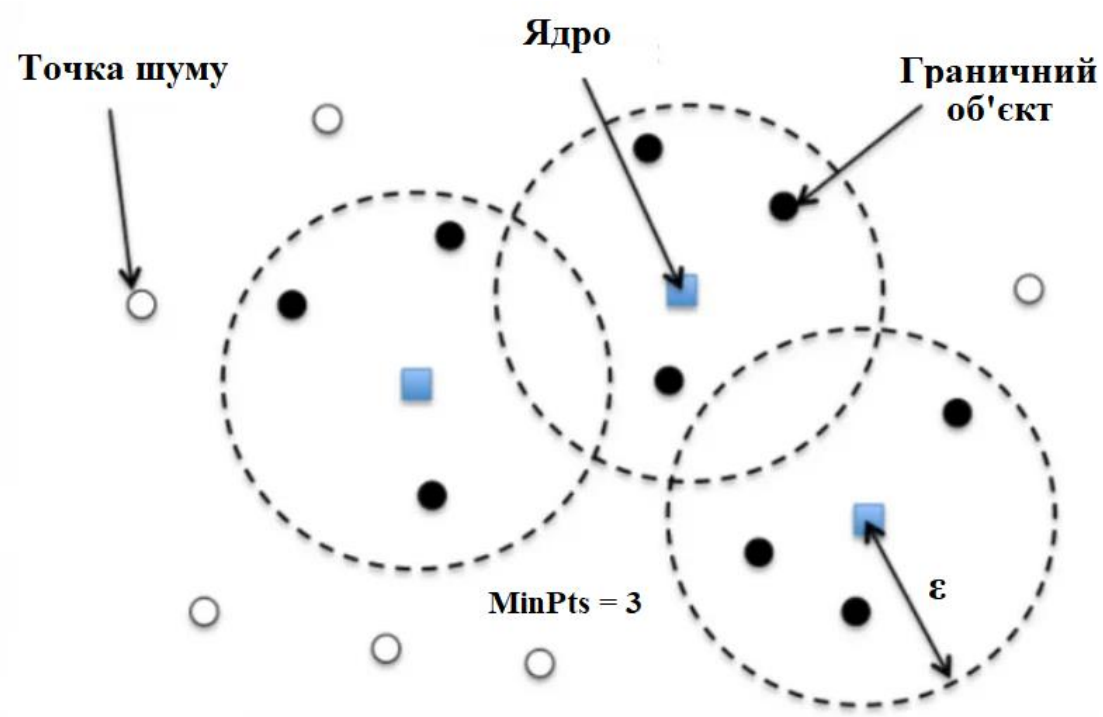

Рисунок 2.4 – Базові концепції алгоритму DBSCAN

Алгоритм кластеризації на основі щільності DBSCAN відмінно підходить для обробки великомасштабних наборів текстів без вказівки кількості класів. Метод має високу швидкодію обробки, ефективно обробляє шумові точки і виявляє кластери довільної форми.

Один із головних мінусів алгоритму полягає у складності вибору найбільш оптимальних значень параметрів, таких як ε та $MinPts$, які можуть варіюватися в залежності від конкретних властивостей даних, їх неправильний вибір може призвести до неправильної кластеризації або навіть повної відсутності кластерів.

Також під час роботи з алгоритмом DBSCAN важливо враховувати, що DBSCAN може виявитися менш ефективним в умовах, коли кластери мають занадто різну щільність, або коли вони мають складну форму.

2.7 Особливості, переваги та недоліки кластеризації на основі векторних представлень слів

Кластеризація на основі векторних представлень слів (Word Embeddings-Based Clustering, далі – WEClustering) – це метод, який використовує векторні представлення слів для групування слів або документів зі схожими семантичними характеристиками. Основна ідея полягає в тому, що семантично схожі слова мають схожі векторні представлення, тому їх можна групувати разом у кластери.

Вектори слів можуть бути отримані за допомогою алгоритмів, таких як Word2Vec, Doc2Vec, GloVe або FastText. За використання цих алгоритмів, слова перетворюються на числові вектори в просторі високої розмірності. Ці вектори захоплюють семантичні відношення між словами, де схожі слова знаходяться близько одне до одного у цьому просторі. Далі до цих векторних представлень застосовують класичні кластерні алгоритми, такі як K-means, ієрархічна кластеризація або DBSCAN.

Основна перевага кластеризації на основі векторних представлень слів полягає в тому, що векторні представлення зберігають семантичні відносини між словами. Таким чином, подібні слова мають схожі векторні представлення і, отже, можуть бути ефективно груповані в кластери. Це дозволяє точно виявляти семантично подібні терміни та концепти в текстових даних. Крім того, врахування контексту допомагає у виявленні семантичних зв’язків, які можуть бути важливими для аналізу тексту. Ще однією перевагою є автоматизація процесу кластеризації, що дозволяє ефективно обробляти великі обсяги текстових даних без необхідності ручної роботи. Цей метод може бути використаний для різноманітних задач, таких як тематичне моделювання, категоризація документів та аналіз семантичних зв’язків між словами [61–63].

З іншого боку, точність кластеризації залежить від якості векторних представлень: якщо вони недостатньо точно відображають семантичні

відносини, результати кластеризації можуть бути неточними. Обчислювальна складність може бути високою, особливо для великих обсягів текстових даних та високо-розмірних векторних просторів. Також інтерпретація отриманих кластерів може бути складною, оскільки їхній зміст не завжди легко розуміти. Отже, вибір параметрів алгоритму кластеризації може впливати на якість результатів, що вимагає додаткового часу та експертизи. Більш того, навчання векторних моделей і їх подальше використання для кластеризації потребує значних обчислювальних ресурсів.

Таким чином, кластеризація на основі векторних представлень слів має значний потенціал для аналізу тексту, зокрема завдяки здатності працювати з контекстом і семантикою. Водночас, її використання вимагає врахування недоліків, таких як статичність моделей та залежність від якості даних.

Більшість сучасних моделей (наприклад, BERT або GPT) вирішують деякі проблеми традиційних word embeddings, застосовуючи контекстно-залежні векторні представлення, що відкриває нові горизонти для кластеризації.

3 ДОСЛІДЖЕННЯ МЕТОДІВ СЕМАНТИЧНОЇ КЛАСТЕРИЗАЦІЇ НОВИН ВІДПОВІДНО ДО ЇХ ТЕМАТИКИ

3.1 Опис методології проведення експериментального порівняльного аналізу

Методологія проведення експериментального порівняльного аналізу полягає у всебічному дослідженні та оцінці ефективності різних алгоритмів кластеризації для групування текстових даних. Це включає кілька етапів, які дозволяють забезпечити надійність та валідність отриманих результатів. У цьому дослідженні буде розглянуто новинний датасет з інформацією про навколишнє середовище, застосовано до нього різні алгоритми кластеризації та порівняно їх ефективність на різних обсягах даних.

Для аналізу було обрано новинний датасет, що містить тексти статей про навколишнє середовище. Він був отриманий з Guardian та містить новини, які необхідно кластеризувати для виявлення основних тем та тенденцій. Вибір датасету зумовлений його релевантністю та багатством змісту, що дозволяє досліджувати різні аспекти кластеризації текстових даних.

Для забезпечення високої якості результатів кластеризації необхідно провести ретельну попередню обробку даних. В даному дослідженні цей етап включає наступні кроки:

Крок 1. Очищення даних. Видалення пропущених значень, стоп-слів, знаків пунктуації та інших небажаних елементів з тексту.

Крок 2. Нормалізація тексту. Перетворення тексту в нижній регістр, лематизація (приведення слів до їх базової форми) та видалення зайвих символів.

Крок 3. Перетворення тексту в числовий формат. Використання методів, таких як TF-IDF (Term Frequency-Inverse Document Frequency), для

перетворення текстових даних у числовий формат, що дозволяє застосовувати алгоритми машинного навчання.

Для порівняння ефективності обрано шість різних методів кластеризації, кожен з яких має свої особливості та застосовується в різних контекстах:

– базовий алгоритм K-Means;

– прихований розподіл Діріхле (LDA);

– Latent Semantic Analysis (LSA) / Latent Semantic Indexing (LSI);

– спектральна кластеризація;

– алгоритм DBSCAN.

Для кожного з обраних алгоритмів буде проведено наступний аналіз:

Крок 1. Застосування алгоритму до початкового датасету. Кластеризація всіх даних для визначення базових характеристик алгоритму.

Крок 2. Застосування алгоритму до скороченого датасету. Кластеризація скороченого датасету для аналізу впливу обсягу даних на результати методу.

Крок 3. Порівняння результатів. Аналіз результатів кластеризації за допомогою метрик, таких як силуетний коефіцієнт, кількість кластерів, час виконання тощо.

Оцінка результатів кластеризації здійснюватиметься за допомогою кількісних та якісних метрик:

– силуетний коефіцієнт – метрика, яка вимірює якість кластеризації, враховуючи середню відстань між точками у кластері та відстань до найближчого кластеру;

– кількість кластерів – оптимальна кількість кластерів визначається для кожного методу, що дозволяє порівняти результати між алгоритмами;

– час виконання – час, необхідний для виконання кластеризації, що дозволяє оцінити ефективність алгоритму з точки зору обчислювальних ресурсів;

– візуалізація кластерів – візуальний аналіз кластерів за допомогою методів зниження розмірності, таких як PCA (Principal Component Analysis) або t-SNE (t-Distributed Stochastic Neighbor Embedding).

Порівняння результатів кластеризації проводиться на основі метрик оцінки та візуального аналізу. Робиться висновок про ефективність кожного методу на різних обсягах даних, визначаються сильні та слабкі сторони кожного алгоритму. Це дозволяє отримати комплексне уявлення про застосування різних методів кластеризації для текстових даних та визначити найбільш підходящі підходи для конкретних завдань.

Ієрархічна кластеризація має високу обчислювальну складність, яка зазвичай оцінюється як O($n^2 \log n$) або O(n^3), де n – кількість об'єктів у датасеті. Для великих текстових датасетів, таких як розглянутий в цьому дослідженні (близько 29,870 текстових документів), обчислення парних відстаней між усіма зразками стає надмірно складним і ресурсномістким. Крім того, потреба у зберіганні матриці відстаней між усіма парами зразків збільшує пам'яткові вимоги, що робить ієрархічну кластеризацію непрактичною для великих обсягів даних.

Текстові дані після перетворення у векторну форму, наприклад, з використанням TF-IDF (Term Frequency-Inverse Document Frequency), зазвичай мають високу розмірність. Високорозмірні дані можуть значно вплинути на продуктивність і точність ієрархічної кластеризації. Методи зменшення розмірності, такі як Principal Component Analysis (PCA), можуть частково вирішити цю проблему, але їх застосування додає ще один рівень складності і не завжди гарантує збереження всіх релевантних інформаційних властивостей даних.

Після побудови дендрограми додавання нових даних або видалення існуючих вимагає повного перерахунку кластеризації. Це робить метод менш адаптивним до динамічно змінюваних датасетів, що є суттєвим недоліком для великих текстових баз даних, які можуть часто оновлюватися. Через високу обчислювальну та пам'яткову складність, чутливість до шуму і викидів, а також відсутність можливості адаптації до змін у датасеті, ієрархічна кластеризація є менш підходящою для великих текстових датасетів. Тому даний метод не було реалізовано.

3.2 Попередня обробка даних

Попередня обробка даних є критично важливим етапом у будь-якому процесі аналізу даних, особливо коли йдеться про текстові дані. Неправильна або неповна обробка може призвести до спотворення результатів і хибних висновків. В цій частині буде розглянуто кроки попередньої обробки даних, які були застосовані до датасету новин Guardian про навколишнє середовище, а також надано більш детальний опис самого датасету.

3.2.1 Опис робочого датасету

Датасет, який використовується для цього дослідження, складається з новинних статей, що стосуються тематики навколишнього середовища. Він містить кілька ключових колонок:

– дата публікації (publication_date) – дата та час, публікації статті;

– заголовок (title) – заголовок новинної статті, яий коротко описує зміст;

– автор (author) – ім’я автора статті;

– текст статті (text) – основний текст, що містить детальний опис події;

– теги (tags) – ключові слова або теги, які описують основні теми статті.

3.2.2 Попередня обробка даних

Щоб забезпечити якість кластеризації, необхідно провести ретельну попередню обробку даних. Нижче описані основні кроки цього процесу.

Крок 1. Очищення даних. Очищення даних включає в себе видалення непотрібної або невірної інформації, яка може негативно вплинути на результати аналізу. Для датасету, що розглядається в рамках цього дослідження, виконуються наступні дії:

– видалення пропущених значень;

– видалення дублікатів.

Крок 2. Нормалізація тексту. Вона допомагає зменшити варіативність даних і підвищити ефективність алгоритмів машинного навчання. Були виконані наступні модифікації:

– перетворення в нижній регістр – всі тексти приводяться до нижнього регістру для уникнення дублювання слів через відмінності у великих і малих літерах;

– видалення знаків пунктуації – з тексту видаляються всі знаки пунктуації, які не несуть семантичного навантаження;

– видалення стоп-слів – таких як "і", "але", "чи", вони не містять значущої інформації для кластеризації;

– лематизація – кожне слово приводиться до його базової форми (леми) для зменшення варіативності та підвищення точності аналізу.

Після очищення та нормалізації тексту необхідно перетворити текстові дані у формат, придатний для аналізу алгоритмами машинного навчання. Для цього використовуватиметься метод TF-IDF (Term Frequency-Inverse Document Frequency), що перетворює текстові дані у числовий формат, що відображає важливість кожного слова у статті відносно всього корпусу документів. TF-IDF дозволяє знизити вагу частих, але малозначущих слів і підвищити вагу рідкісних, але важливих термінів. Після TF-IDF векторизації створюється матриця ознак, де кожен рядок відповідає одній статті, а кожен стовпчик – одній ознаці (терміну). Ця матриця використовується як вхідні дані для алгоритмів кластеризації.

3.2.3 Результати попередньої обробки датасету

Процес проведення попередньої обробки датасету, обраного для проведення досліджень відображено на рисунку 3.1.

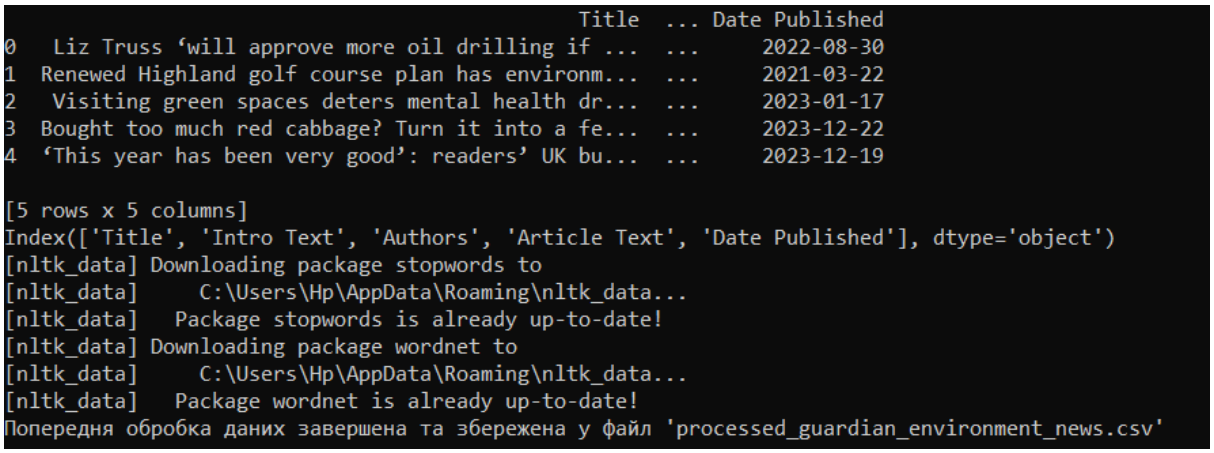

Рисунок 3.1 – Результати попередньої обробки датасету

Наступним кроком є реалізація методів семантичної кластеризації для скороченого і повного датасетів.

3.3 Програмна реалізація та візуалізація результатів виконання методів семантичної кластеризації

3.3.1 Метод K-Means

Першим було реалізовано алгоритм K-Means. Він є одним із найпоширеніших методів кластеризації, що використовується для розділення даних на k кластерів. Кожен кластер визначається своїм центром (центроїдом), і кожна точка даних належить до кластера з найближчим центроїдом. Слід пам’ятати, що алгоритм є доволі чутливий до початкового вибору центроїдів, що може призводити до різних результатів при різних запусках.

На рисунку 3.2 наведено блок-схему алгоритму, за якою було здійснено програмну реалізацію методу K-Means.

Логіку методу можна описати наступним чином:

– ініціалізація k центроїдів випадковим чином;

– кожна точка даних призначається до кластера, центроїд якого знаходиться найближче до неї за певною метрикою (зазвичай використовується евклідова відстань);

– повторювати до збіжності призначення кожної точки до найближчого центроїда та обчислення нових центроїдів як середнього значення точок, призначених до кожного кластера.

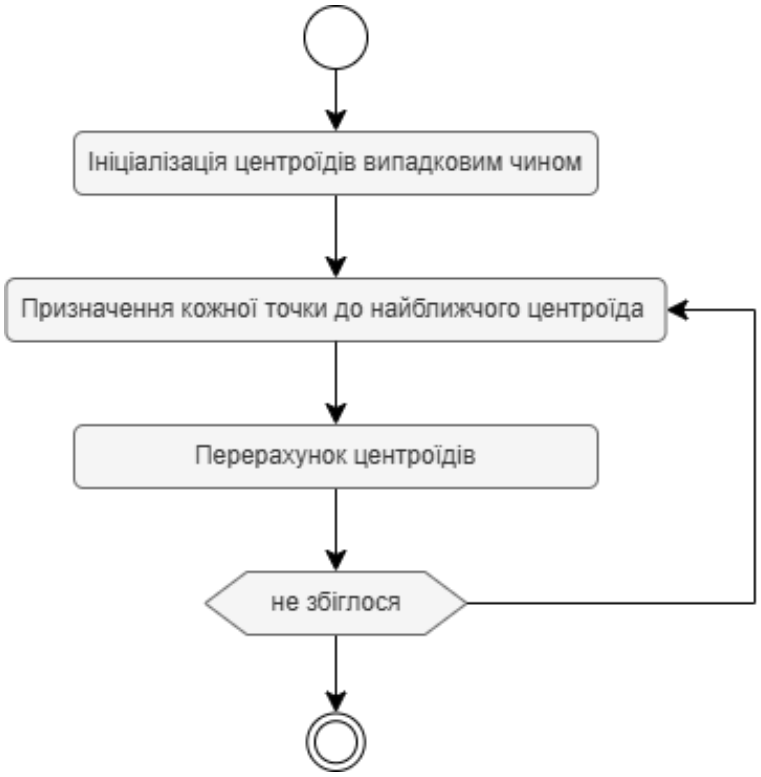

Рисунок 3.2 – Блок-схема алгоритму K-Means

Параметри, використані під час програмної реалізації алгоритму K-Means виглядають наступним чином:

– n_clusters – кількість кластерів k;

– random_state – ініціалізація з метою забезпечення відтворюваності результатів;

– max_iter – максимальна кількість ітерацій алгоритму;

– n_init – кількість різних ініціалізацій алгоритму.

Після відпрацювання створених програм, слід візуалізувати результати та здійснити їх аналіз. Тестування проводилось на повному датасеті та скороченому на 50%.

Детальні результати семантичної кластеризації за методом K-means наведені на рисунку 3.3 та на рисунку 3.4.

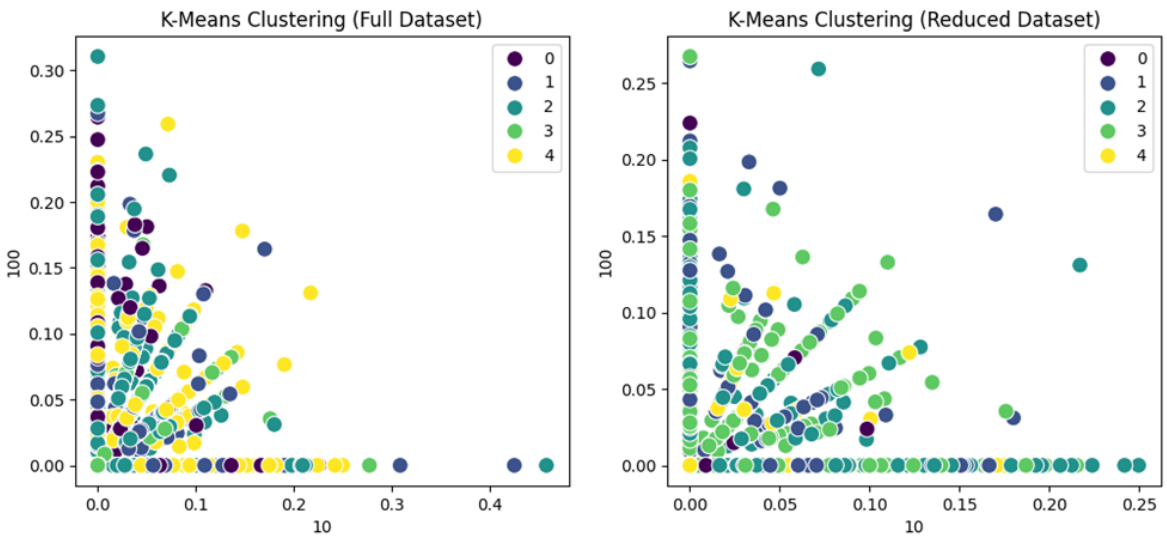

Рисунок 3.3 – Результати кластеризації для методу K-means

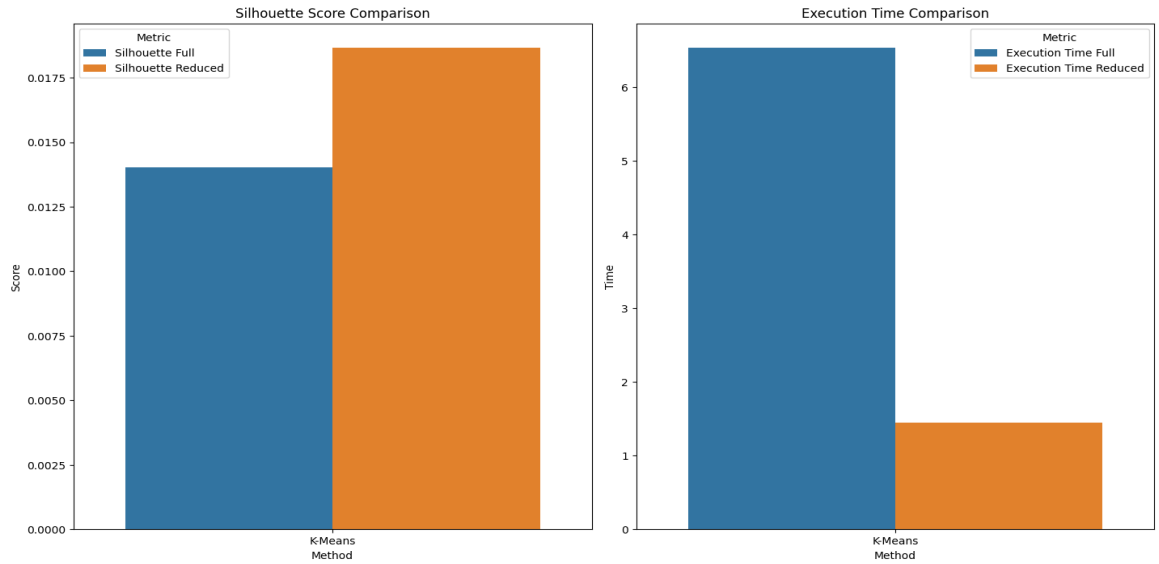

Рисунок 3.4 – Оцінка кластеризації для методу K-means

Графіки відображають кластери та оцінки, за ними можна відслідкувати, що результати кластеризації відрізняються в залежності від розміру датасету.

Силуетний коефіцієнт є показником якості кластеризації. Він вищий для скороченого датасету, що може вказувати на кращу роздільність кластерів при меншій кількості даних. В результаті відпрацювання алгоритму K-means силуетний коефіцієнт дорівнює наступним показникам: приблизно 0,014 для повного датасету та приблизно 0, 018 для скороченого. При цьому час

виконання становив 4,24 та 1,08 секунди для повного та скороченого датасетів відповідно.

Кластеризація на скороченому датасеті показала трохи кращі результати за силуетним коефіцієнтом, такий підхід може бути корисним для попереднього аналізу даних або коли швидкість виконання є критично важливою. Час виконання суттєво зменшується при скороченні датасету, що підтверджує, що для великих обсягів даних оптимізація розміру датасету може значно прискорити аналіз.

3.3.2 Метод LDA

Наступний реалізований алгоритм, LDA (Latent Dirichlet Allocation), є тематичним моделлю, яка використовується для знаходження прихованих тем у текстових даних. Вона розглядає документи як суміш тем, а теми як суміш слів.

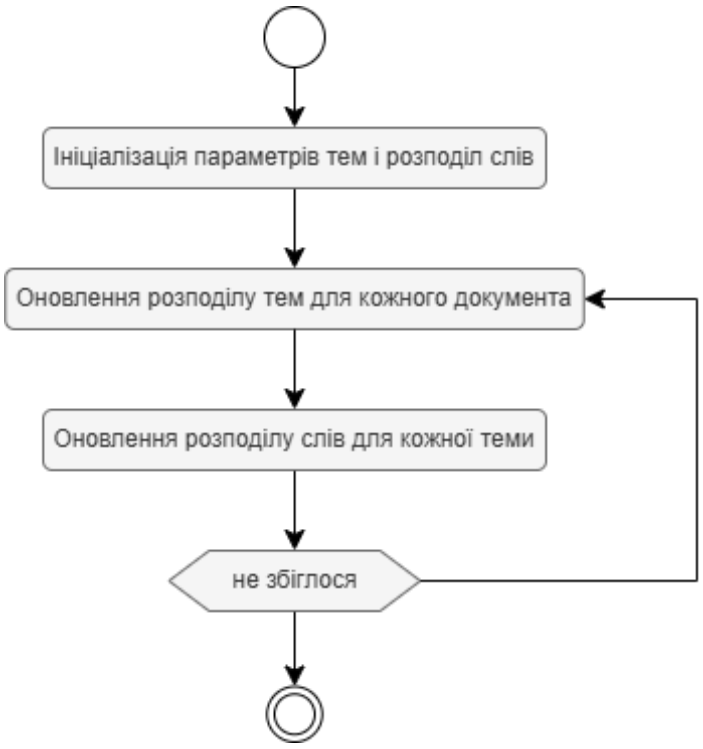

Рисунок 3.5 – Блок-схема алгоритму LDA

Логіка методу виглядає наступним чином:

– ініціалізація параметрів тем і розподілів слів;

– оновлення розподілу тем для кожного з текстових документів;

– оновлення розподілу слів для кожної теми;

– повторення процесу до досягнення збіжності, при якій зміни в розподілах між ітераціями стають незначними.

Для реалізації LDA було використано такі параметри, як:

– n_components – кількість тем для знаходження;

– max_iter – максимальна кількість ітерацій алгоритму;

– learning_method – метод навчання ('batch' / 'online').

На рисунках 3.6-3.7 відображено результати виконання кластеризації за методом LDA.

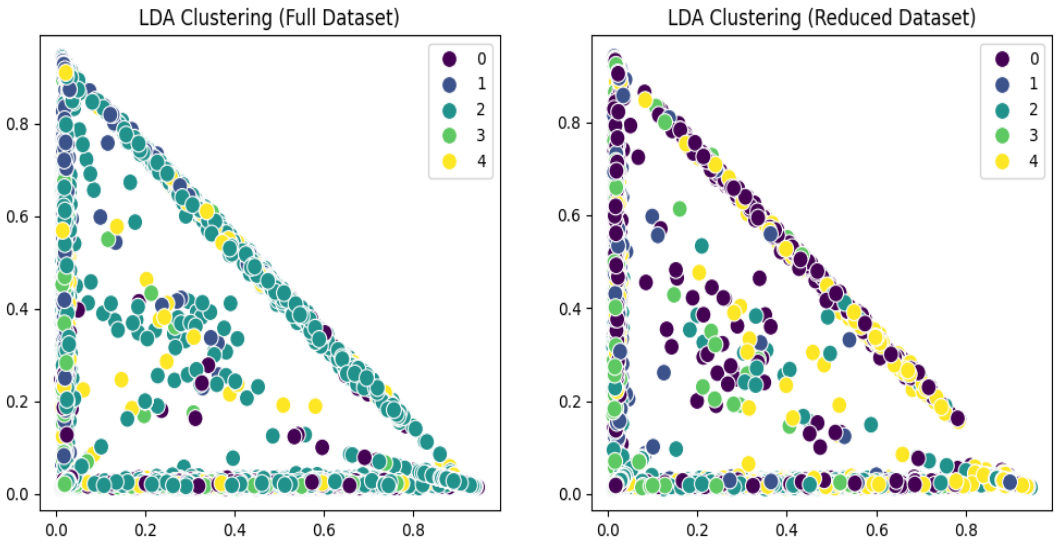

Рисунок 3.6 – Результати кластеризації за методом LDA

З рисунку 3.6 видно, що при застосуванні LDA до повного датасету (ліва частина) точки розподілені більш рівномірно, але існує значна змішаність між кластерами. Для зменшеного датасету спостерігається дещо більша концентрація точок у певних областях, що може вказувати на краще відокремлення кластерів.

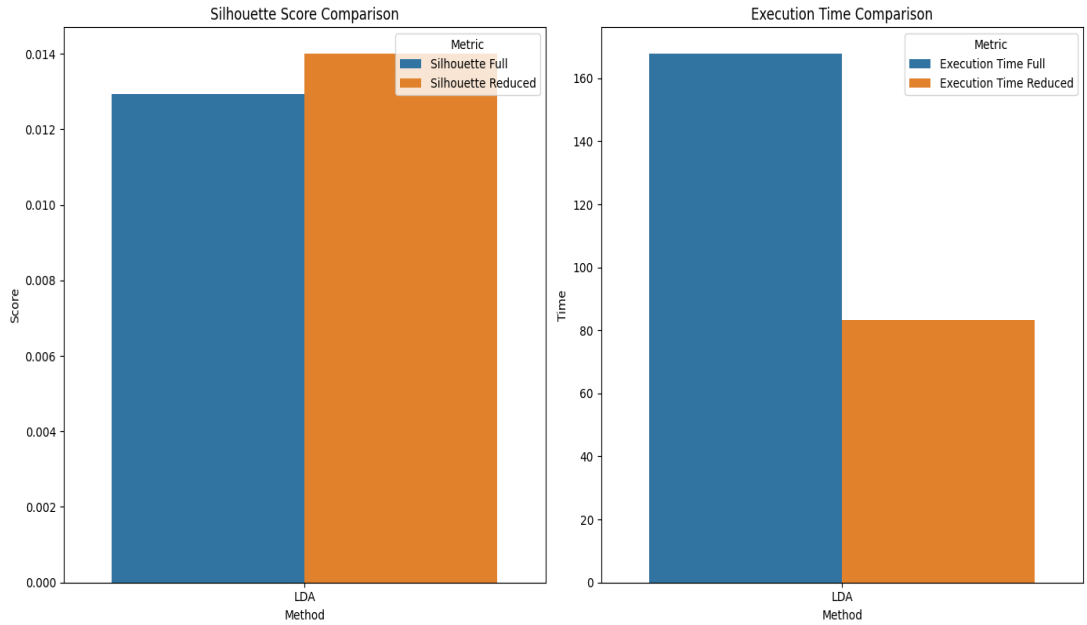

Рисунок 3.7 – Оцінка для методу LDA

За результатами кластеризації, силуетний коефіцієнт склав приблизно 0,0129 для повного датасету та 0,0139 для скороченого. Час виконання кластеризації сягав 167,7 секунд для повного датасету та 83,40 секунд для скороченого.

Зменшений датасет має трохи вищий силуетний коефіцієнт, що вказує на дещо кращу кластеризацію. Це може бути пов’язано з тим, що менший обсяг даних дозволяє моделі краще визначати межі кластерів. Час виконання для повного датасету значно більший порівняно із зменшеним, що є очікуваним результатом і підкреслює важливість зменшення розміру датасету для зменшення витрат на обчислення.

3.3.3 Метод LSA

LSA використовує сингулярне розкладання для зменшення розмірності текстових даних, виявляючи приховані семантичні структури (рис. 3.8). Для цього використовується сингулярне розкладання матриці (Singular Value

Decomposition, SVD), що зменшує розмірність даних і виділяє основні семантичні компоненти. Метод складається з обчислення сингулярного розкладання для матриці термінів-документів та вибору кількох найбільших сингулярних значень для побудови оновленної зменшеної матриці.

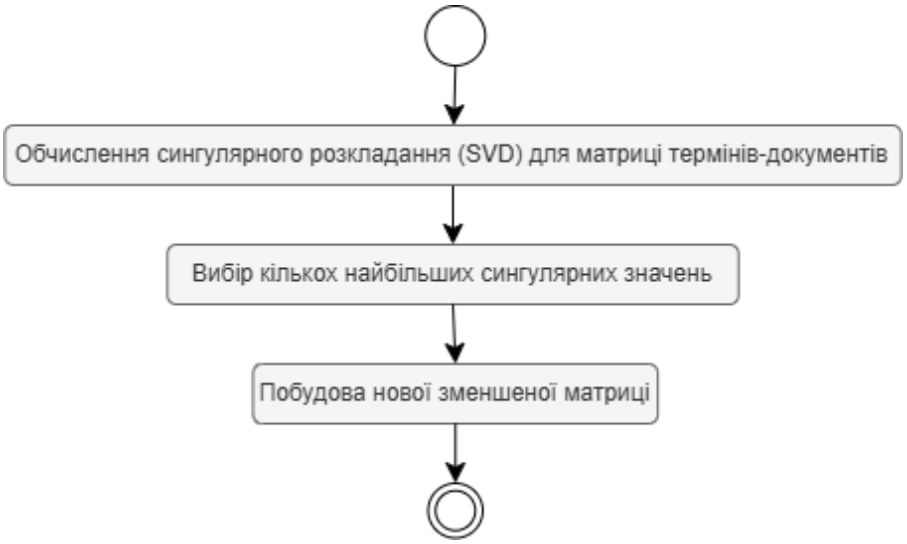

Рисунок 3.8 – Блок-схема алгоритму LSA

Параметри, створені під час реалізації:

– n_components – кількість компонент для зменшення розмірності;

– algorithm – метод обчислення SVD ('randomized', 'arpack').

Результати кластеризації, здійсненої за алгоритмом алгоритму LSA наведено на рисунку 3.9.

На рисунку 3.9 зображено візуалізацію кластерів після відпрацювання програми за LSA. Повний датасет має чітко відображені кластери, з різними кольорами для кожного кластеру. При роботі зі скороченим датасетом кластери також чітко відображені, але структура може бути менш виражена у порівнянні з повним датасетом.

На рисунку 3.10 можна спостерігати порівняння силуетного коефіцієнта та часу виконання для повного та скороченого датасетів.

Силуетний коефіцієнт для повного датасету сягнув приблизно 0,0025, а для скороченого датасету – 0,0015. Це означає, що LSA працює краще з

повним датасетом, оскільки силуетний коефіцієнт вищий, об’єкти кластерів більш схожі між собою і менш схожі з об’єктами інших кластерів.

Час виконання кластеризації методом LSA для повного датасету складає 1,0 секунд, а для скороченого датасету приблизно 0,8 секунд. Час виконання трохи зменшується при використанні скороченого датасету, що логічно, оскільки обробляється менша кількість даних.

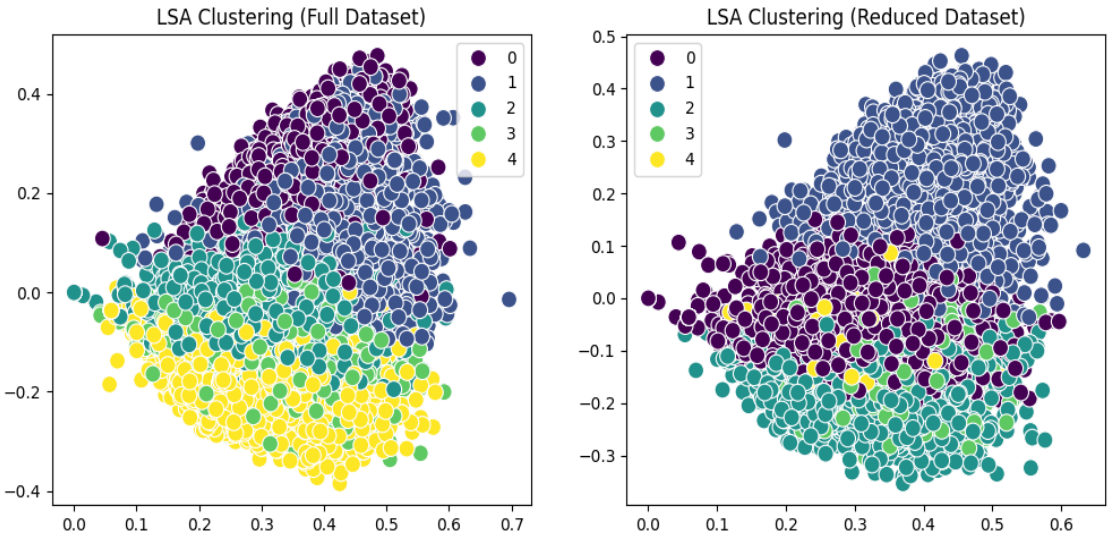

Рисунок 3.9 – Результати кластеризації методом LSA

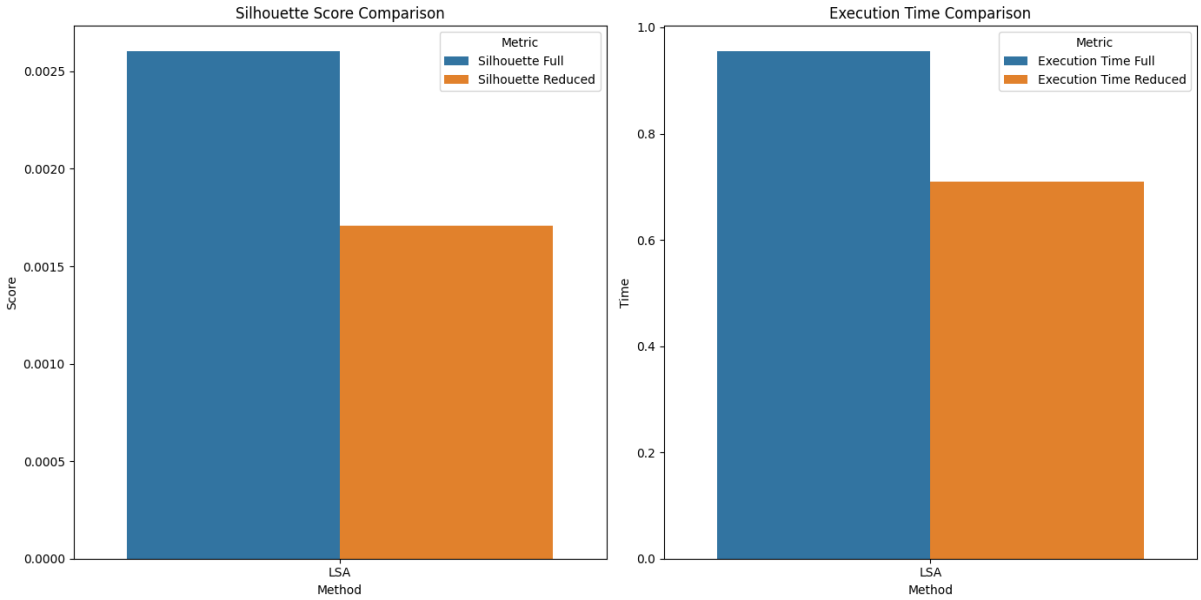

Рисунок 3.10 – Оцінка для методу LSA

3.3.4 Метод Spectral Clustering

Spectral Clustering використовує спектральні (власні) значення матриці відстаней для виконання кластеризації. Цей метод підходить для виявлення складних нелінійних зв’язків між даними (рис.3.11).

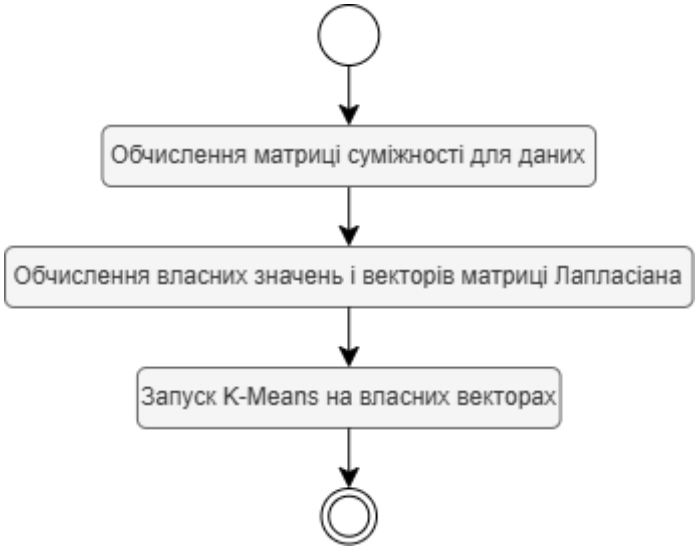

Рисунок 3.11 – Блок-схема алгоритму Spectral Clustering

Логіка Spectral Clustering складається з наступних етапів:

– обчислення матриці суміжності для даних;

– обчислення власних значень і власних векторів матриці Лапласіана;

– виконання кластеризації (K-Means) на власних векторах.

Параметри, які були використані під час програмної реалізації:

– n_clusters – кількість кластерів;

– affinity – спосіб обчислення суміжності (наприклад, 'nearest_neighbors', 'rbf');

– n_init – кількість різних ініціалізацій алгоритму K-Means.

Результати роботи алгоритму Spectral показано на рисунку 3.12. На обох графіках (ліворуч для повного датасету, а праворуч – для зменшеного) видно, що точки кластерів сильно змішуються. Більшість точок мають однаковий кластерний індекс (0), що вказує на слабку сегментацію даних.

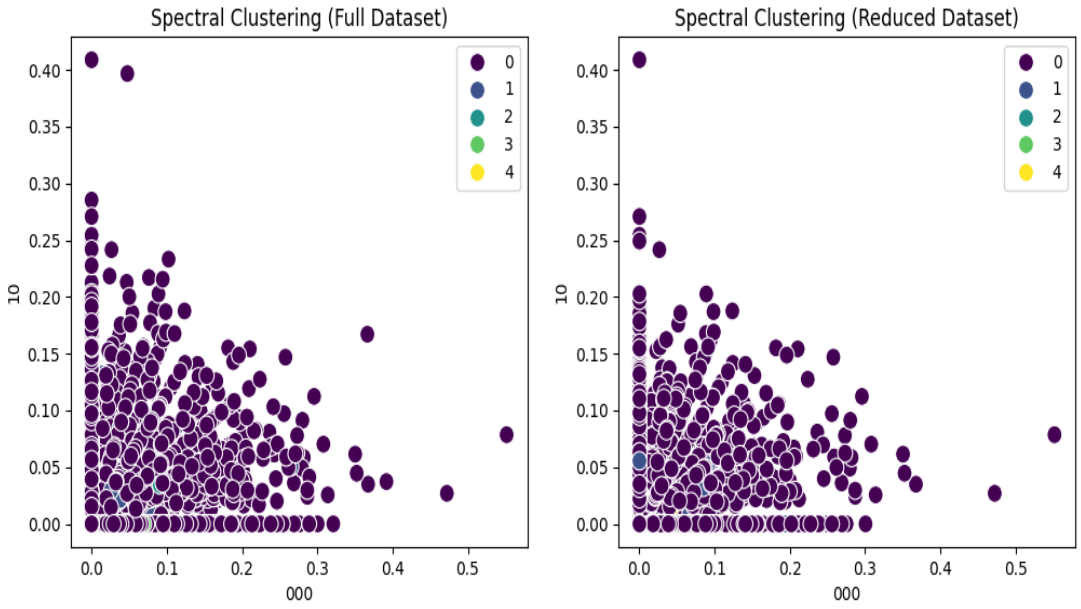

Рисунок 3.12 – Результати для методу Spectral

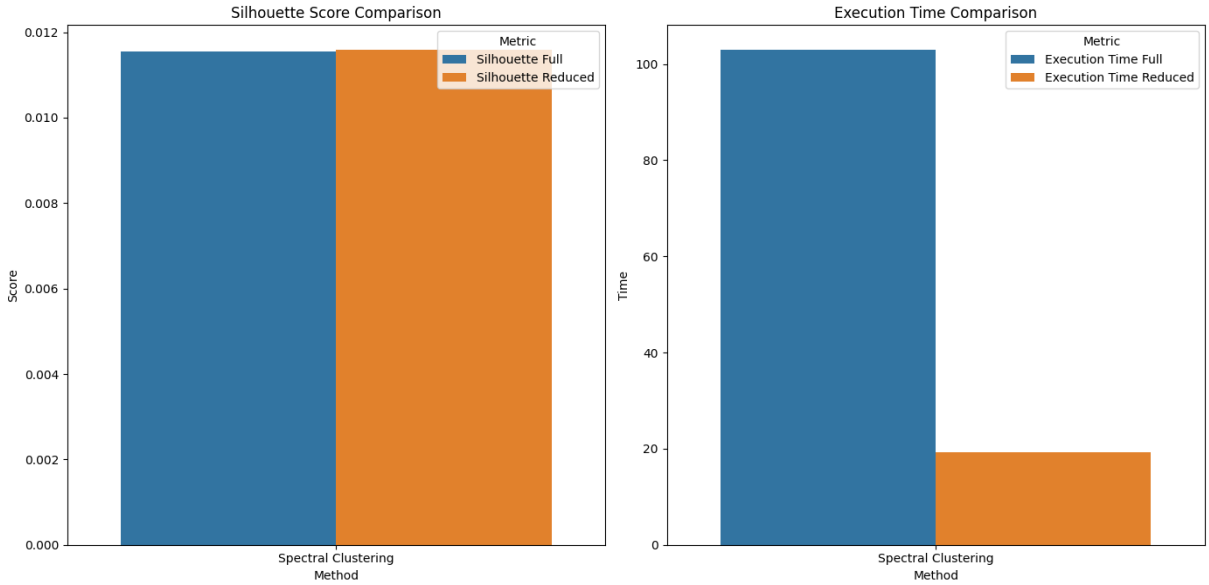

Рисунок 3.13 – Оцінка для методу Spectral

Візуалізація оцінка методу зображено на рисунку 3.13. Силуетний коефіцієнт для повного датасету становить приблизно 0,01195533 за час виконання 104,34 секунд, а силуетний коефіцієнт для скороченого датасету – 0,01198454 за 52,28 секунд. Значення силуетного коефіцієнта для обох датасетів дуже близькі і невеликі, що вказує на незначну якість кластеризації.

Час виконання для повного датасету значно більший порівняно із зменшеним датасетом, що є очікуваним результатом. Силуетний коефіцієнт для обох випадків дуже низький (близько 0,012), що вказує на те, що спектральна кластеризація не змогла ефективно сегментувати дані.

3.3.5 Метод DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) є методом кластеризації на основі щільності, який визначає кластери як області високої щільності, розділені областями низької щільності (рис. 3.14). Він також виявляє шуми в даних.

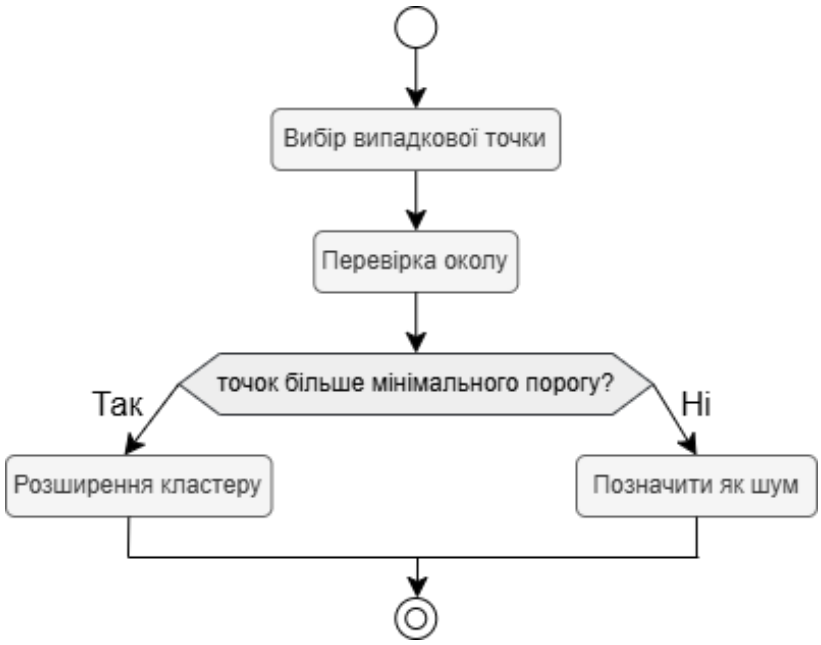

Рисунок 3.14 – Блок-схема алгоритму DBSCAN

Логіка алгоритму DBSCAN:

– вибір випадкової точки та перевірка її околу;

– якщо кількість точок в околі перевищує мінімальний поріг, точка стає центром кластера;

– усі точки в околі приєднуються до кластера, процес повторюється;

– якщо точка не входить в жоден кластер, вона вважається шумом.

Для програмної реалізації алгоритму DBSCAN було використано такі параметри, як:

– eps – максимальна відстань між точками для їхнього включення в кластер;

– min_samples – мінімальна кількість точок, необхідних для формування кластера;

– metric – метрична функція для обчислення відстаней.

На рисунках 3.15 та 3.16 зображено результати кластеризації датасету за методом DBSCAN.

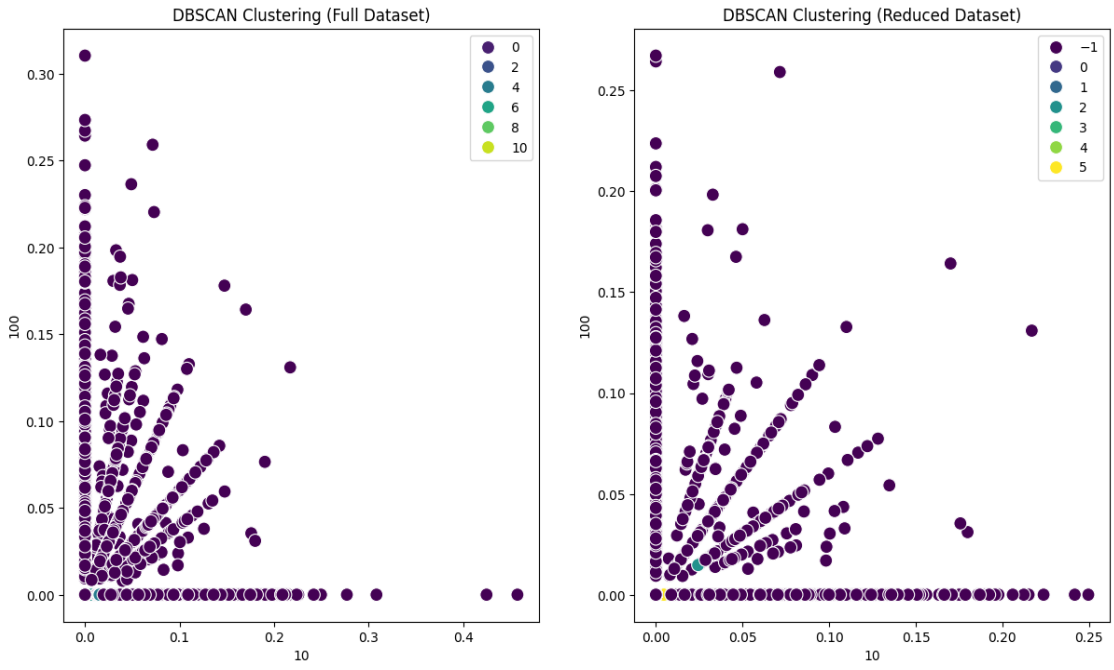

Рисунок 3.15 – Результати кластеризації за DBSCAN

Велика кількість точок має значення "0", це показує, що багато точок не були кластеризовані (були віднесені до шуму).

З рисунку 3.16 видно, що значення силуетного коефіцієнту під час кластеризації за DBSCAN сягнули 0,643 за умови використання повного датасету та 0,738 – за скороченого. Таким чином, DBSCAN працює краще зі скороченим датасетом, оскільки силуетний коефіцієнт вищий.

Час виконання кластеризації займав для повного датасету приблизно 42,88 секунди, а для скороченого – 13,25 секунд.

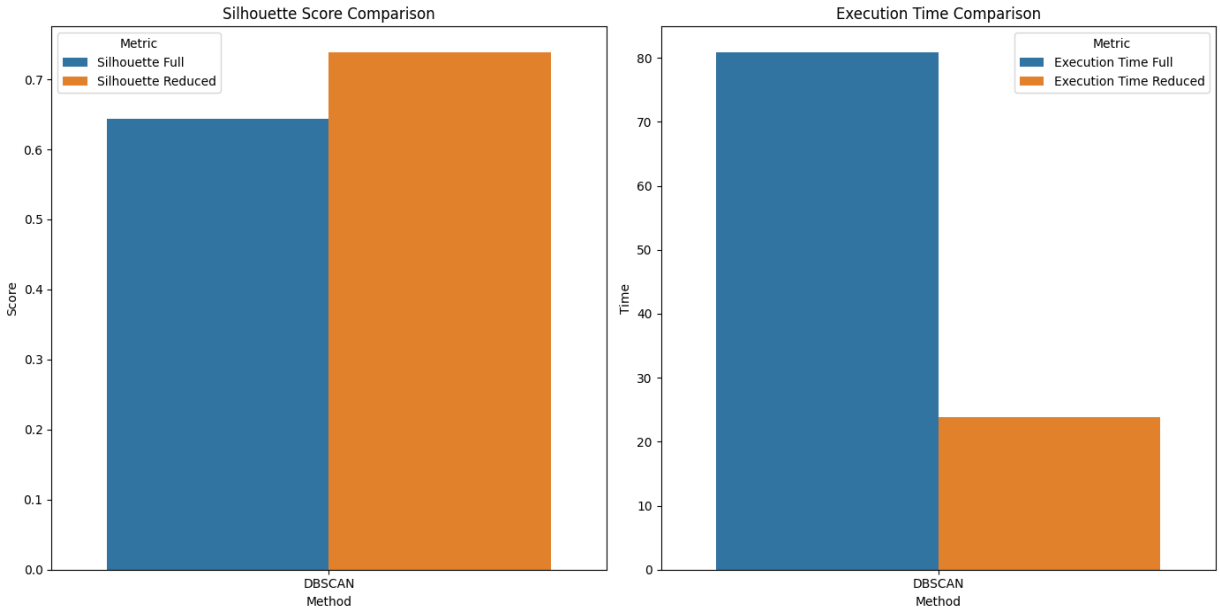

Рисунок 3.16 – Оцінка кластеризації методом DBSCAN

3.4 Порівняння отриманих результатів

Підведемо підсумки реалізації алгоритмів семантичної кластеризації новин відповідно до їх тематики. В таблиці 3.1 наведено розгорнутий порівняльний аналіз результатів відпрацювання програм.

Дослідження підтвердило, що скорочення розміру датасету позитивно впливає на час виконання алгоритмів, що є важливим для обчислювально інтенсивних методів, таких як LDA та Spectral Clustering. Проте якість кластеризації за силуетним коефіцієнтом не завжди корелює зі скороченням даних. Це свідчить, що для певних методів (наприклад, DBSCAN) якість значно залежить від структурних характеристик вибірки. Показник силуетного коефіцієнту варіюється залежно від обраного методу кластеризації, деякі з них мають низькі значення на обох наборах, що свідчить про слабку сегментацію.

Таблиця 3.1 – Порівняння методів

Метод	**Силуетний коефіцієнт (повний)**	**Силуетний коефіцієнт (скорочений)**	**Час виконання (повний, с)**	**Час виконання (скорочений, с)**
K-Means	0,014	0,018	4,24	1,08
LDA	0,0129	0,0139	167,7	83,40
LSA	0,0025	0,0015	1,0	0,8
Spectral Clustering	0,01195533	0,01198454	104,34	52,28
DBSCAN	0,643	0,738	42,88	13,25

Методи K-Means та LDA показали найкращу якість кластеризації, в особливості для меншого за розміром датасету, при чому час виконання кластеризації методом LDA суттєво більший. Spectral Clustering та DBSCAN не змогли ефективно сегментувати дані: не дивлячись на значення силуетних коефіцієнтів, більшість точок були не кластеризовані, а віднесені до шуму. LSA відпрацював краще на вибірці з більшою кількістю даних, однак різниця у часі виконання між повним і скороченим датасетами не є суттєвою.

ВИСНОВКИ

У рамках кваліфікаційної роботи було проведено дослідження та програмний порівняльний аналіз п’ятьох методів семантичної кластеризації, таких як K-Means, LDA, LSA, Spectral Clustering та DBSCAN.

За результатами реалізації методів семантичної кластеризації для кластеризації новинних даних можна зробити наступні висновки. Оптимізація розміру датасету значно впливає як на точність кластеризації, так і на обчислювальні ресурси, необхідні для виконання алгоритмів. Метод K-Means продемонстрував стабільні результати щодо якості кластеризації при меншому часі виконання, що робить його ще ефективнішим для задач із меншими обсягами даних. Метод LDA також показав непогані результати, однак його час виконання є значно більшим, що обмежує його застосування для великих наборів даних [64].

Spectral Clustering та DBSCAN виявилися менш ефективними у кластеризації новинних матеріалів. Хоча DBSCAN продемонстрував високий силуетний коефіцієнт, значна частина точок була класифікована як шум, що свідчить про недостатню точність методу. Метод LSA виявився придатним для обробки великих обсягів даних, однак його ефективність при зменшенні розміру датасету залишилась на низькому рівні. Таким чином, результати дослідження підтверджують, що вибір методу кластеризації та розмір датасету мають вирішальне значення для якості аналізу новинних даних.

Наукова новизна роботи полягає у оптимізації алгоритмів кластеризації для текстових даних, що дозволяє підвищити точність тематичного групування новин і масштабованість для роботи з великими обсягами даних.

Подальші перспективи дослідження у сфері семантичної кластеризації новин можуть бути спрямовані на вдосконалення вже існуючих алгоритмів та розробку нових підходів для покращення точності та ефективності кластеризації.

По-перше, варто розглянути можливість використання комбінованих або гібридних методів кластеризації, які об’єднують сильні сторони декількох алгоритмів для підвищення точності та скорочення часу виконання. Наприклад, комбінація K-Means із більш просунутими методологіями, такими як глибоке навчання чи нейронні мережі. Такий підхід може покращити не лише якість кластеризації, а й зменшити обчислювальні витрати, особливо при роботі з великими обсягами даних.

По-друге, подальші дослідження можуть бути зосереджені на адаптації алгоритмів кластеризації до динамічних змін у новинних потоках. Тобто потрібно розробити методи, які зможуть не лише ефективно кластеризувати новини, але й швидко адаптуватися до появи нових тем або змін у популярності існуючих.

Результати дослідження апробовано у вигляді тез доповідей під час 28-го Міжнародного молодіжного форуму «РАДІОЕЛЕКТРОНІКА І МОЛОДЬ У XXI СТОЛІТТІ» [65].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Garcia-Dias, R., Vieira, S., Pinaya, W. H. L., & Mechelli, A. (2020). Clustering analysis. *In machine learning* (pp. 227-247). Academic Press.

2. Kobylin, O., Vyskrebentseva, S., & Petrova, R. (2019). Обробка даних, що містять пропуски в задачах кластеризації. *Системи управління, навігації та зв'язку. Збірник наукових праць*, *5*(57).

3. Naik, M. P., Prajapati, H. B., & Dabhi, V. K. (2015, March). A survey on semantic document clustering. In *2015 IEEE international conference on electrical, computer and communication technologies (ICECCT)* (pp. 1-10). IEEE.

4. Bodyanskiy, Y., Kobylin, I., Rashkevych, Y., Vynokurova, O., & Peleshko, D. (2018, February). Hybrid fuzzy-clustering algorithm of unevenly and asynchronously spaced time series in computer engineering. In *2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET)* (pp. 930-935). IEEE.

5. Mashtalir, V., Ruban, I., & Levashenko, V. (Eds.). (2019). Advances in Spatio-Temporal Segmentation of Visual Data (Vol. 876). Springer Nature.

6. Antons, D., Grünwald, E., Cichy, P., & Salge, T. O. (2020). The application of text mining methods in innovation research: current state, evolution patterns, and development priorities. *R&D Management*, 50(3), 329-351.

7. Kobylin, O. A., Gorokhovatskyi, V. O., Tvoroshenko, I. S., & Peredrii, O. O. (2020). The application of non-parametric statistics methods in image classifiers based on structural description components. Telecommunications and Radio Engineering, 79(10).

8. Talib, R., Hanif, M. K., Ayesha, S., & Fatima, F. (2019). Text mining: techniques, applications and issues. *International journal of advanced computer science and applications*, 7(11), 414-418.

9. Gorokhovatskyi, V., & Tvoroshenko, I. (2023). Identification of visual objects by the search request. *International scientific symposium «INTELLIGENT SOLUTIONS-S»* (pp. 25-27).

10. Liu, Y.-C., Liu, M., & Wang, X.-L. (2020). Application of Self-Organizing Maps in Text Clustering: A Review. InTech, 5-35.

11. Mashtalir, S., Mashtalir, V., & Stolbovyi, M. (2018, August). Representative Based Clustering of Long Multivariate Sequences with Different Lengths. In *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)* (pp. 545-548). IEEE.

12. Selvaraj S, Choi E. Dynamic Sub-Swarm Approach of PSO Algorithms for Text Document Clustering. *Sensors*. 2022; 22(24):9653.

13. Ahmed MH, Tiun S, Omar N, Sani NS. Short Text Clustering Algorithms, Application and Challenges: A Survey. *Applied Sciences*. 2023; 13(1):342.

14. Daradkeh, Y. I., Gorokhovatskyi, V., Tvoroshenko, I., Gadetska, S., & Al-Dhaifallah, M. (2023). Statistical data analysis models for determining the relevance of structural image descriptions. IEEE Access, 11, 126938-126949.

15. Palanivinayagam, A.; Nagarajan, S. An optimized iterative clustering framework for recognizing speech. *Int. J. Speech Technol*. 2020, 23, 767–777.

16. Gorokhovatskyi V., Tvoroshenko I., Kobylin O., and Vlasenko N. (2023) Search for visual objects by request in the form of a cluster representation for the structural image description, Advances in Electrical and Electronic Engineering, 21(1), pp. 19-27.

17. Gorokhovatskyi V., Tvoroshenko I., Kobylin O., and Vlasenko N. (2023) Search for visual objects by request in the form of a cluster representation for the structural image description, Advances in Electrical and Electronic Engineering, 21(1), pp. 19-27.

18. Rosid, M. A., Fitrani, A. S., Astutik, I. R. I., Mulloh, N. I., & Gozali, H. A. (2020, June). Improving text preprocessing for student complaint document classification using sastrawi. In *IOP Conference Series: Materials Science and Engineering* (Vol. 874, No. 1, p. 012017). IOP Publishing.

19. Anandarajan, M., Hill, C., Nolan, T., Anandarajan, M., Hill, C., & Nolan, T. (2019). Text preprocessing. *Practical text analytics: Maximizing the value of text data*, 45-59.

20. Kobylin, O. A., Gorokhovatskyi, V. O., Tvoroshenko, I. S., & Peredrii, O. O. (2020). The application of non-parametric statistics methods in image classifiers based on structural description components. Telecommunications and Radio Engineering, 79(10).

21. Almuzaini, H. A., & Azmi, A. M. (2020). Impact of stemming and word embedding on deep learning-based Arabic text categorization. *IEEE Access*, 8, 127913-127928.

22. Ibrahim, D. Y., Gorokhovatskyi, V., Tvoroshenko, I., & Mujahed, A. D. (2022). Classification of Images Based on a System of Hierarchical Features.

23. Pal, T. L., Kumari, M., Singh, T., & Ahsan, M. (2018). Semantic representations in text data. *International Journal of Grid and Distributed Computing*, 11(9), 65-80.

24. Rabotiahov, A., Kobylin, O., Dudar, Z., & Lyashenko, V. (2018, February). Bionic image segmentation of cytology samples method. In *2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET)* (pp. 665-670). IEEE.

25. Gupta, A., Kumar, A., Gautam, J., Gupta, A., Kumar, M. A., & Gautam, J. (2017). A survey on semantic similarity measures. *International Journal for Innovative Research in Science & Technology*, 3(12), 243-247.

26. Kobylin, O., & Lyashenko, V. (2016). Contrast Modification as a Tool to Study the Structure of Blood Components.

27. Prasetya, D. D., Wibawa, A. P., & Hirashima, T. (2018). The performance of text similarity algorithms. *International Journal of Advances in Intelligent Informatics*, 4(1), 63-69.

28. Kinoshenko, D., Kobylin, O., Mashtalir, S., & Stolbovyi, M. (2019, March). Metric video retrieval speedup by irrelevant data elimination. In Eleventh International Conference on Machine Vision (ICMV 2018) (Vol. 11041, pp. 176-183). SPIE.

29. Prakoso, D. W., Abdi, A., & Amrit, C. (2021). Short text similarity measurement methods: a review. *Soft Computing*, 25, 4699-4723.

30. Tvoroshenko, I. S., & Bielinskyi, Y. (2021). On the features of methods of processing and recognition of handwritten text.

31. Varghese, N., & Punithavalli, M. (2020). Semantic similarity analysis on knowledge-based and prediction based models. *Int J Innov Technol Exploring Eng*, 9(6), 447-481.

32. Gorokhovatskyi, V. A., Rusakova, N., & Tvoroshenko, I. S. (2020). The application of image analysis methods and predicate logic in applied problems of magnetic monitoring. Telecommunications and Radio Engineering, 79(20).

33. Daradkeh, Y. I., & Tvoroshenko, I. (2020). Technologies for making reliable decisions on a variety of effective factors using fuzzy logic. International Journal of Advanced Computer Science and Applications, 11(5).

34. HE, Z., DUMDUMAYA, C. E., & QUIMNO, V. (2024). MEASUREMENT OF SEMANTIC TEXT SIMILARITY. *Journal of Theoretical and Applied Information Technology*, 102(5).

35. Tvoroshenko, I., Ahmad, M. A., Mustafa, S. K., Lyashenko, V., & Alharbi, A. R. (2020). Modification of models intensive development ontologies by fuzzy logic.

36. Salih, N. M., & Jacksi, K. (2020, December). Semantic Document Clustering using K-means algorithm and Ward's Method. In 2020 *International Conference on Advanced Science and Engineering* (ICOASE) (pp. 1-6). IEEE.

37. Lyashenko, V., Matarneh, R., Kobylin, O., & Putyatin, Y. (2016). Contour detection and allocation for cytological images using Wavelet analysis methodology.

38. Yang, J., & Wang, J. (2017). Tag clustering algorithm LMMSK: improved K-means algorithm based on latent semantic analysis. *Journal of Systems Engineering and Electronics*, 28(2), 374-384.

39. Moseley, B., & Wang, J. R. (2023). Approximation bounds for hierarchical clustering: Average linkage, bisecting k-means, and local search. *Journal of Machine Learning Research*, 24(1), 1-36.

40. Gorokhovatskyi, V. O., Tvoroshenko, I. S., & Vlasenko, N. V. (2020). Using fuzzy clustering in structural methods of image classification. Telecommunications and Radio Engineering, 79(9).

41. Bisecting Kmeans Clustering. URL: https://medium.com/@afrizalfir/bisecting-kmeans-clustering-5bc17603b8a2 (дата звернення 05.10.2024).

42. Yakovleva, O., & Nikolaieva, K. (2020). Research of descriptor based image normalization and comparative analysis of SURF, SIFT, BRISK, ORB, KAZE, AKAZE descriptors. Advanced Information Systems, 4(4), 89-101.

43. Xu, J., Zhao, T., Feng, G., Ni, M., & Ou, S. (2021). A fuzzy C-means clustering algorithm based on spatial context model for image segmentation. *International Journal of Fuzzy Systems*, 23, 816-832.

44. Daradkeh, Y. I., & Tvoroshenko, I. (2020). Technologies for making reliable decisions on a variety of effective factors using fuzzy logic. International Journal of Advanced Computer Science and Applications, 11(5).

45. Pomazan, V., Tvoroshenko, I., & Gorokhovatskyi, V. (2023). Development of an application for recognizing emotions using convolutional neural networks. *International Journal of Academic Information Systems Research*, 7(7), (pp. 25-36).

46. Lyashenko V., Kobylin O., Selevko O. (2020) Wavelet Analysis and Contrast Modification in the Study of Cell Structures Images. International Journal of Advanced Trends in Computer Science and Engineering. 9(4). – 4701-4706.

47. Krasnov, D., Davis, D., Malott, K., Chen, Y., Shi, X., & Wong, A. (2023). Fuzzy c-means clustering: A review of applications in breast cancer detection. *Entropy*, 25(7), 1021.

48. Reddy, C. K., & Vinzamuri, B. (2018). A survey of partitional and hierarchical clustering algorithms. In Data clustering (pp. 87-110). Chapman and Hall/CRC.

49. Oleg, K., Sergii, M., & Mykhailo, S. (2017, October). Video Clustering via Multidimensional Time-Series Analysis. In *Proceedings of the 9th International Conference on Information Management and Engineering* (pp. 60-63). ACM.

50. Understanding the concept of Hierarchical clustering Technique. URL: https://towardsdatascience.com/understanding-the-concept-of-hierarchical-clustering-technique-c6e8243758ec (дата звернення 15.10.2024).

51. Hierarchical Clustering in Data Mining. URL: https://www.geeksforgeeks.org/hierarchical-clustering-in-data-mining/ (дата звернення 15.10.2024)

52. Кобилін, О. А., & Творошенко, І. С. (2021). Методи цифрової обробки зображень.

53. Haji, S. H., Jacksi, K., & Salah, R. M. (2023). A Semantics-Based Clustering Approach for Online Laboratories Using K-Means and HAC Algorithms. *Mathematics*, 11(3), 548.

54. Lyashenko, V., Mohammad, A., & Kobylin, O. (2015). Experiments with Fusion of Images with Use of Wavelet Transformation in Problems of the Text Information Analysis.

55. Liu, Z. (2013). *High performance latent dirichlet allocation for text mining* (Doctoral dissertation, Brunel University School of Engineering and Design PhD Theses).

56. Is Latent Dirichlet Allocation (LDA) A Clustering Algorithm? URL: https://highdemandskills.com/lda-clustering/ (дата звернення 20.10.2024)

57. Yakovleva, O., Kovtunenko, A., Liubchenko, V., Honcharenko, V., & Kobylin, O. (2023). Face Detection for Video Surveillance-based Security System. In COLINS (3) (pp. 69-86).

58. Yakovleva, O., Slyusar, V., Kushnir, O., & Sabovchyk, A. (2021). New trends in scientific and technological revolution (STR) and transformation of science and education systems in the paradigm of sustainable development. In E3S Web of Conferences (Vol. 277, p. 06006). EDP Sciences.

59. Latent Dirichlet Allocation. URL: https://medium.com/@corymaklin/latent-dirichlet-allocation-dfcea0b1fddc (дата звернення 20.10.2024).

60. Spectral Clustering in Machine Learning. URL: https://www.geeksforgeeks.org/ml-spectral-clustering/ (дата звернення 25.10.2024).

61. Bodyanskiy, Y., Vynokurova, O., Kobylin, I., & Kobylin, O. (2016). Adaptive fuzzy clustering of short time series with unevenly distributed observations in Data Stream Mining tasks. *Information Technology and Management Science*, *19*(1), 23-28.

62. Gorokhovatskyi, V., Tvoroshenko, I., & Olena, Y. (2024). Transforming image descriptions as a set of descriptors to construct classification features. *Indonesian Journal of Electrical Engineering and Computer Scienc*, 33 (1), (pp. 113-125)

63. Kuzminska, O., Mazorchuk, M., Morze, N., & Kobylin, O. (2019, June). Digital learning environment of ukrainian universities: The main components to influence the competence of students and teachers. In International Conference on Information and Communication Technologies in Education, Research, and Industrial Applications (pp. 210-230). Springer, Cham.

64. Кобилін, О. А., & Путятіна, О. Є. (2024). Знешумлення зображень, зіпсованих дробовим шумом, у реальному часі. Системи обробки інформації, (1 (176), 46-51.

65. Гончарова О.В. (2024) Дослідження методів семантичної кластеризації для аналізу новин // Радіоелектроніка та молодь у XXI столітті. Т.7: Конференція «Комп'ютерний зір, системний аналіз та математичне Моделювання»: матеріали 28-го Міжнар. молодіж. форуму, 16–18 квіт. 2024 р. / М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Харків: ХНУРЕ, 2024. – 320 с. – DOI: 10.30837/IYF.CVSAMM.2024. – с. 31-32.