Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження та застосування методів NLP для вирішення проблеми холодного старту в рекомендаційних системах
(тема)

Виконав:
студент 2 курсу, групи СШІм-22-3
Грішаєва А.М.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма Системи штучного інтелекту
(повна назва спеціалізації)
Керівник проф. Рябова Н.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ В.О. Філатов
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

Рівень вищої освіти другий (магістерський)

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

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Грішаєвій Анастасії Миколаївні
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та застосування методів NLP для вирішення проблеми холодного старту в рекомендаційних системах

затверджена наказом університету від 1 квітня 20 24 р. № 260Ст
2. Термін подання студентом роботи до екзаменаційної комісії 5 червня 20 24 р.
3. Вихідні дані до роботи Науково-технічні публікації, дані Інтернет-джерел та відомих наукових проектів, Python документація, набір даних для тренування та тестування системи

4. Перелік питань, що потрібно опрацювати в роботі
1) Аналіз предметної галузі
2) Опис проведених теоретичних досліджень
3) Опис системи, що пропонується
4) Опис проведених експериментальних досліджень

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	01.04.2024	виконано
2	Аналіз предметної галузі	15.04.2023	виконано
3	Визначення проблеми холодного старту	18.04.2024	виконано
4	Роль NLP у вирішенні проблеми	23.04.2024	виконано
5	Програмна реалізація	30.04.2024	виконано
6	Аналіз результатів	05.05.2024	виконано
7	Написання пояснювальної записки	10.05.2024	виконано
8	Перевірка на академічний плагіат	25.05.2024	виконано
9	Нормоконтроль	27.05.2024	виконано
10	Підготовка презентації та доповіді	30.05.2024	виконано
11	Попередній захист	02.06.2024	виконано
12	Рецензування	03.06.2024	виконано
13	Захист перед ЕК	05.06.2024	

Дата видачі завдання 1 квітня 20 24 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Рябова Н.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 68 с., 17 рис., 2 табл., 2 дод., 21 джерело.

ВЕКТОРНЕ ПРЕДСТАВЛЕННЯ ТЕКСТІВ, КЛАСТЕРИЗАЦІЯ, РЕКОМЕНДАЦІЙНИ СИСТЕМИ, ХОЛОДНИЙ СТАРТ, NATURAL LANGUAGE PROCESSING.

Об’єкт дослідження – проблема холодного старту у рекомендаційних системах.

Предмет дослідження – методи та алгоритми обробки природньої мови для вирішення проблеми холодного старту.

Мета роботи полягає у розробці та аналізі алгоритму рекомендаційної системи, який ефективно вирішує проблему холодного старту за допомогою застосування методів обробки природної мови (NLP), для покращення персоналізації та точності рекомендацій у різних сферах, зокрема в сфері технічної і наукової літератури.

Методи дослідження – аналітичний (аналіз існуючих даних, теорій та підходів), експериментальний (розробка та тестування алгоритмів рекомендаційної системи, заснованих на NLP, оцінка їх ефективності).

Розроблено алгоритм надання рекомендацій у випадку холодного старту, який може використовуватись як стартова точка для запуску системи. Використання методів обробки природньої мови дозволило провести контент-аналіз та максимально зменшити час обробки і отримання рекомендацій.

ABSTRACT

Master’s thesis contains: 68 pp., 17 fig., 2 tabl., 2 ann., 21 references.

CLUSTERING, COLD START, NATURAL LANGUAGE PROCESSING, RECOMMENDATION SYSTEMS, TEXT VECTOR REPRESENTATION.

The object of research is the cold start problem in recommendation systems.

The subject of research is methods and algorithms of natural language processing for solving the cold start problem.

The work aims to develop and analyze a recommendation system algorithm that effectively addresses the cold start problem using natural language processing (NLP) methods, to improve personalization and accuracy of recommendations in various fields, including the domain of technical and scientific literature.

Research methods include analytical (analysis of existing data, theories, and approaches), experimental (development and testing of recommendation system algorithms based on NLP, assessment of their effectiveness).

An algorithm for providing recommendations in the case of a cold start has been developed, which can be used as a starting point for launching the system. The use of natural language processing methods has allowed for content analysis and has minimized the time required for processing and obtaining recommendations.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів 8
Вступ 9
1 Аналіз предметної галузі 11
1.1 Загальне введення в область рекомендаційних систем та їх важливість у сучасному інтернет-просторі 11
1.2. Рекомендаційні моделі 12
1.2.1 Колаборативна фільтрація 13
1.2.2 Контент-базова фільтрація 14
1.2.3 Гібридні системи 16
1.3. Роз'яснення терміну «холодний старт» та пояснення виникнення проблеми 17
1.4 Розгляд існуючих способів вирішення проблеми 18
1.4.1 Загальні підходи надання рекомендацій 20
1.5 Оцінки та метрики подібності 21
1.6 Етичні аспекти та приватність даних 25
1.7 Предметні області використання NLP для вирішення проблеми холодного старту 26
1.8 Постановка задачі 29
1.9 Висновки до розділу 29
2 Вибір засобів вирішення задачі та розробка алгоритму 31
2.1 Розгляд методів NLP 31
2.2 Крос-модальне навчання в системах NLP 33
2.3 Вибір сфери та напрямку NLP 35
2.4 Семантичні методи 36
2.5 Методи векторного представлення 37
2.5.1 Модель «Мішок слів» 37
2.5.2 Модель Word2Vec 38
2.5.3 GloVe 39

2.6 Методи кластеризації 39
2.6.1 K-means кластеризація 39
2.6.2 Ієрархічна кластеризація 41
2.7 Розробка алгоритму 42
2.8 Висновки до розділу 45
3 Програмна реалізація 46
3.1 Вибір датасету 46
3.2 Опис датасету 47
3.3 Аналіз та підготовка даних 48
3.4 Використання Word2Vec 51
3.5 Кластеризація 54
3.6 Аналіз результатів 58
3.7 Висновки до розділу 60
Висновки 61
Перелік джерел посилання 62
Додаток А Програмний код 64
Додаток Б Відомість кваліфікаційної роботи 68

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ШІ – штучний інтелект;

BERT – Bidirectional Encoder Representations from Transformers – двонаправлені представлення енкодерів від трансформерів;

CBOW – Continuous Bag of Words – неперервний мішок слів;

CLIP – Contrastive Language-Image Pre-training – контрастивне переднавчання мови та зображень;

GAN – Generative Adversarial Network – генеративні змагальні мережі;

GloVe – Global Vectors for Word Representation -– глобальні вектори для представлення слів;

GPT – Generative Pre-trained Transformer – генеративний переднавчений трансформер;

LDA – Latent Dirichlet Allocation – латентне розподілення Діріхле;

NLP – Natural Language Processing – обробка природньої мови;

ViT – Vision Transformer – трансформер зображень.

ВСТУП

У сучасному цифровому світі, де обсяги даних зростають експоненційно, рекомендаційні системи набувають все більшого значення у поліпшенні користувацького досвіду, сприяючи ефективному відбору контенту. Проблема холодного старту – це ключовий виклик у цій області, який полягає у вирішенні завдання надання персоналізованих рекомендацій новим користувачам або для нових об'єктів без історії взаємодій. Ця проблема обмежує ефективність рекомендаційних систем, особливо на початкових етапах їх застосування.

Розвиток методів обробки природної мови (NLP) відкриває нові можливості для вирішення проблеми холодного старту, дозволяючи системам краще розуміти контент і користувацькі запити. Актуальність цієї роботи полягає у пошуку ефективних методів NLP, які можуть бути застосовані для поліпшення точності та персоналізації рекомендацій, що сприятиме зростанню користувацького задоволення та вірності.

Ціль роботи полягає у дослідженні та розробці алгоритму на основі NLP для вирішення проблеми холодного старту в рекомендаційних системах. Можливі сфери застосування включають електронну комерцію, онлайн-освіту, стрімінгові платформи для перегляду відео та слухання музики, соціальні мережі, де існує потреба в ефективній рекомендаційній системі.

У першому розділі ми розглянемо теоретичні основи рекомендаційних систем, включаючи колаборативну фільтрацію, контент-базову фільтрацію та гібридні моделі.

Далі ми детально розглянемо проблему холодного старту, аналізуючи її причини та наслідки для користувачів та платформ. Будуть представлені існуючі підходи до її вирішення – як з використання NLP-методів, так і без них.

У другому розділі ми зосередимося на виборі засобів вирішення

поставленої задачі та розробці алгоритму, який ефективно впорається з нею, зокрема в контексті обробки природної мови. Ми детально розглянемо різноманітні методи NLP, їх переваги та обмеження, щоб зрозуміти, які з них найкраще підходять для нашої конкретної задачі. Особлива увага буде приділена різним методам векторного представлення текстів та методам кластеризації.

У третьому розділі розглядатиметься практична реалізація обраних методів NLP для розв'язання проблеми холодного старту. Ми детально описуємо процес вибору, налаштування та тестування алгоритмів, включаючи підготовку даних, векторизацію текстів та застосування алгоритмів кластеризації. Окрему увагу буде приділено аналізу ефективності та точності рекомендацій, отриманих за допомогою розробленої системи, а також обговоренню її потенційних областей застосування та впливу на користувацький досвід.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Загальне введення в область рекомендаційних систем та їх важливість у сучасному інтернет-просторі

В епоху цифрової інформації, де інтернет-простір переповнений даними, рекомендаційні системи є не лише інструментом, але й необхідністю, яка допомагає користувачам навігувати крізь море контенту. Ці системи є важливими для підтримки уваги та взаємодії користувачів, адаптуючись до їх унікальних інтересів та потреб [1].

Рекомендаційні системи втілюють собою сукупність складних алгоритмів, які аналізують поведінку користувачів, їх попередні вподобання та інтеракції, використовуючи методики такі як колаборативне фільтрування, фільтрування на основі контенту, глибоке навчання, та інші штучно-інтелектуальні технології [2]. Ці системи не просто спрощують пошук контенту, але й здатні передбачати та антиципувати потреби користувачів, часто розкриваючи їм контент, про який вони можливо навіть не знали.

З постійно збільшуючимся обсягом доступного контенту, завдання вибору стає все більш важким. Рекомендаційні системи відіграють ключову роль у фільтрації та персоналізації інформації, що дає змогу користувачам отримувати релевантний і цікавий контент без непотрібного перевантаження інформацією [3]. Вони дозволяють користувачам відкривати новий контент, що не тільки відповідає їхнім смакам, але й стимулює відкриття нових інтересів та знань, забезпечуючи освітній та розважальний досвід.

Крім того, рекомендаційні системи несуть значну вартість для власників контенту та платформ. Вони не тільки збільшують залученість користувачів та час, проведений на платформі, але й сприяють зростанню конверсії та доходів, рекомендуючи продукти або послуги, які мають більшу ймовірність придбання.

Враховуючи широкий спектр застосування – від електронної комерції до стрімінгових сервісів та соціальних мереж – рекомендаційні системи стали неодмінною частиною бізнес-стратегій, спрямованих на підвищення задоволеності клієнтів та оптимізацію їхнього досвіду. Ці системи допомагають утримати користувачів на платформі, забезпечуючи їм постійний потік відповідного контенту, та стимулюють повторні візити та покупки.

Інтелектуальні рекомендаційні алгоритми, які застосовують передові технології, включаючи машинне навчання та NLP, здатні глибше аналізувати не тільки поведінку користувачів, але й їхній зворотний зв'язок, сентименти та вподобання, які виявляються у відгуках та взаємодії з контентом. Це дає можливість створювати більш точні профілі інтересів, що, в свою чергу, призводить до підвищення лояльності клієнтів та їхньої готовності рекомендувати сервіси іншим.

Актуальність рекомендаційних систем сьогодні також проявляється в їхній здатності допомагати компаніям адаптуватися до змінюваних тенденцій ринку та змін у споживацьких перевагах. Вони стають інструментом для збору цінної аналітичної інформації, яка може впливати на стратегічне планування, асортимент продуктів, маркетингові кампанії та навіть розробку нових продуктів.

В результаті, рекомендаційні системи не просто полегшують навігацію по інтернет-простору для користувачів, але й стають стратегічним активом для бізнесів, що шукають шляхи для підвищення конкурентоспроможності та інноваційності у динамічному цифровому світі.

1.2. Рекомендаційні моделі

Рекомендаційні системи можуть бути класифіковані залежно від типу даних, які вони аналізують, і від способів, якими вони формують рекомендації. Від персоналізованих до неперсоналізованих, ці підходи

допомагають формувати користувацький досвід, надаючи контент, що найкраще відповідає індивідуальним перевагам [3].

1.2.1 Колаборативна фільтрація

Цей метод розраховує на спільність переваг серед користувачів або елементів. Він використовує історію інтеракцій користувачів для визначення, які елементи вони можуть оцінити високо. Цей метод є персоналізованим.

Колаборативна фільтрація генерує рекомендації на основі подібності між користувацькими профілями або між елементами, що раніше сподобалися користувачу [4].

Діаграма колаборативної фільтрації надана на рисунку 1.1.

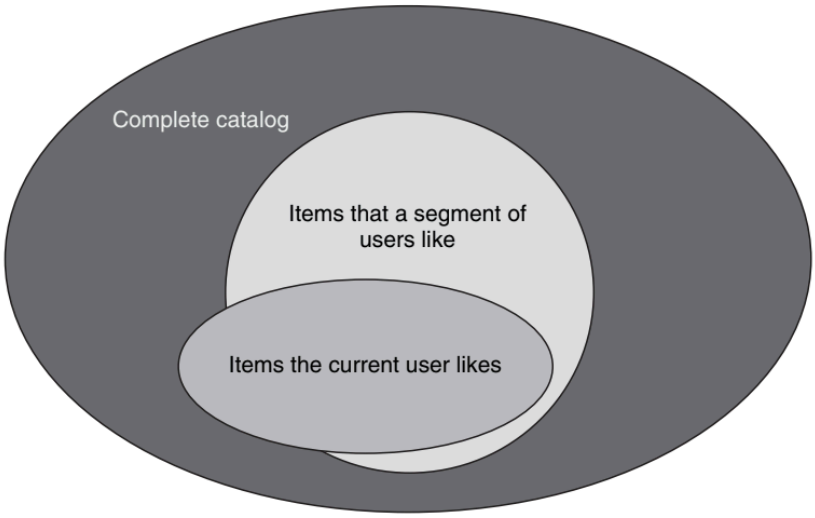

Рисунок 1.1 – Діаграма колаборативної фільтрації

Переваги та недоліки колаборативної фільтрації наведені у таблиці 1.1.

Таблиця 1.1 – Переваги та недоліки колаборативної фільтрації

Переваги	Недоліки
Ефективно враховує унікальні переваги користувачів, надаючи високо персоналізовані рекомендації.	Важко надавати рекомендації новим користувачам або для нових елементів без історії взаємодій.
Може виявити неочікувані інтереси користувача або рекомендувати елементи, на які користувач не натрапив би самостійно.	Вимагає великої кількості даних про взаємодії, а в системах з великою кількістю елементів це може стати проблемою.
Використовує «мудрість натовпу» для покращення якості рекомендацій, виходячи з поведінкових моделей ширшої аудиторії.	Алгоритми, які шукають схожих користувачів або елементи, можуть бути обчислювально вимогливими при великих наборах даних.

1.2.2 Контент-базова фільтрація

Цей підхід аналізує атрибути елементів, які користувач вже оцінив, та рекомендує нові елементи зі схожими характеристиками. Цей метод може бути персоналізованим та неперсоналізованим.

Персоналізована контент-базова фільтрація використовує детальний аналіз контенту, який користувач вже полюбляє, для створення індивідуальних рекомендацій, оптимізованих під його унікальний смак.

Неперсоналізована контент-базова фільтрація може рекомендувати елементи на основі їх загальної популярності або актуальності, не враховуючи індивідуальних переваг. Наприклад, можна рекомендувати 10 найпопулярніших фільмів на даний момент.

Контент-базова фільтрація є ефективним методом у випадках, коли

важливо рекомендувати контент, який має специфічні властивості або коли користувачі мають чіткі вподобання щодо характеристик контенту. Переваги та недоліки можна побачити у таблиці 1.2.

Таблиця 1.2 – Переваги та недоліки контент-базової фільтрації

Переваги	Недоліки
Рекомендації базуються на характеристиках елементів, що робить метод ефективним навіть при обмеженій кількості користувачів.	Спирається на наявність та якість метаданих або описів елементів, що може обмежувати його ефективність.
Легше інтерпретувати та пояснити, чому певний елемент був рекомендований, оскільки він базується на конкретних характеристиках.	Схильний до рекомендації елементів, що лише близькі до вже оцінених користувачем, знижуючи вірогідність відкриття чогось справді нового.
Може рекомендувати нові елементи без історії взаємодій, оскільки оцінює їх засновано на характеристиках.	Важко адаптувати рекомендації для нових користувачів, оскільки системі потрібно спочатку зібрати інформацію про їхні інтереси.

Для вирішення проблеми рекомендації лише близьких елементів можна скористуватись наступним алгоритмом, який представлено на рисунку 1.2.

Для цього необхідно визначити, до якої категорії (назвемо її А) належить елемент. Після цього ми визначаємо найближчу категорію до категорії А (нехай це буде категорія Б). Тепер ми можемо рекомендувати елементи з категорії Б – наприклад, найпопулярніший, або обрати інший алгоритм (знайти найближчий елемент, тощо) відповідно до потреб нашої предметної області.

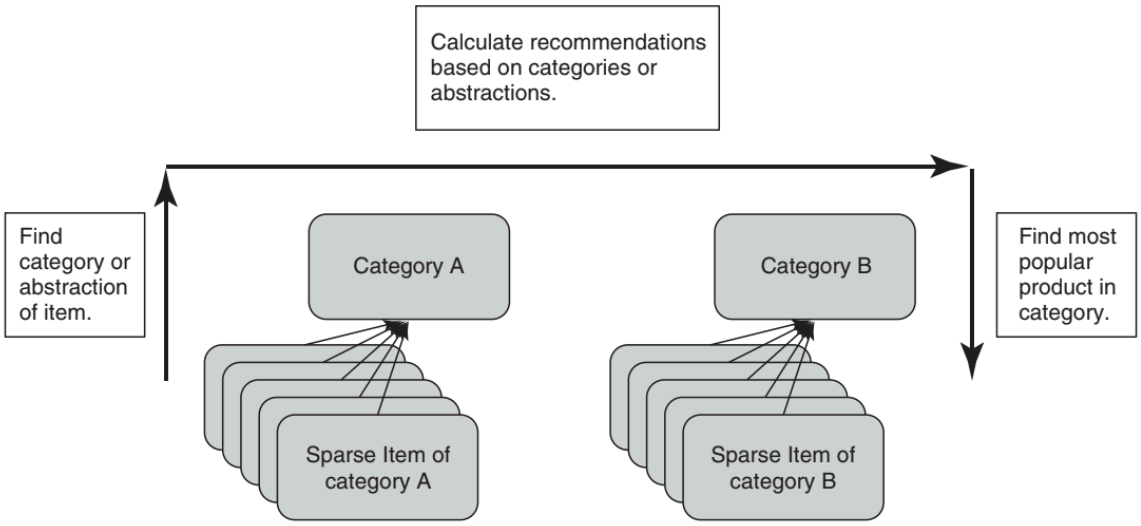

Рисунок 1.2 – Вирішення проблеми рекомендації лише схожих елементів

Абстракція або класифікація не може бути занадто загальною, оскільки втрачається цінність асоціації між категоріями. Прикладом такої асоціації у жанрах фільмів є екшн => комедія. Можна знайти багато асоціацій у даних, але мало цінності в обчисленні якісних рекомендацій.

1.2.3 Гібридні системи

Комбінація обох підходів, як правило, пропонує більш балансовані та точні рекомендації, використовуючи переваги кожного методу для заповнення прогалин іншого.

Гібридні системи інтегрують дані з колаборативної фільтрації та контент-базової фільтрації для створення комплексного профілю користувача, забезпечуючи глибоко персоналізовані рекомендації.

Також гібридні системи можуть використовувати різні методи в залежності від кількості даних користувача. Таким чином, на початку використовуються методи контент-базової фільтрації, а після того, як

отримана необхідна кількість даних, починають використовуватися методи колаборативної фільтрації.

В цілому, успіх рекомендаційної системи залежить від її здатності точно розуміти та передбачати інтереси користувачів, враховуючи як їхні прямі дії, так і більш тонкі, неявні сигнали. Персоналізація є ключовим фактором у забезпеченні задоволення та залученості користувачів, тоді як неперсоналізовані підходи можуть бути ефективними для нових користувачів або коли дані для персоналізації [6].

1.3. Роз'яснення терміну «холодний старт» та пояснення виникнення проблеми

Термін «холодний старт» у контексті рекомендаційних систем відноситься до ситуації, коли система зіштовхується з викликом надання персоналізованих рекомендацій без попередніх даних про взаємодію [3]. Ця проблема виникає в двох основних сценаріях:

Нові користувачі. Коли користувач вперше реєструється на платформі, відсутність історії переглядів та попередніх взаємодій залишає рекомендаційну систему без необхідного контексту для створення персоналізованих пропозицій. Це може призвести до загальних та нецільових рекомендацій, які не відповідають індивідуальним інтересам користувача, потенційно знижуючи його задоволеність та залученість [7].

Нові об'єкти. Аналогічно, коли нові об'єкти (товари, фільми, статті тощо) додаються до системи, вони не мають історії взаємодії, що ускладнює задачу рекомендаційної системи щодо ідентифікації та пропозиції цих об'єктів відповідним користувачам. Без зворотного зв'язку від користувачів, система не може ефективно оцінити потенційну важливість або популярність цих нових елементів.

Проблема холодного старту є фундаментальною, оскільки вона стосується основоположних принципів рекомендаційних систем, а саме

здатності адаптуватися та вчитися з взаємодій користувачів. Ця проблема має як технічні, так і практичні аспекти, враховуючи, що вона впливає на користувацький досвід з моменту першого взаємодії з системою.

Інноваційні підходи до розв'язання проблеми холодного старту включають використання демографічних даних для створення початкових гіпотез про інтереси користувача, експлуатацію метаданих для рекомендації нових об'єктів на основі їх атрибутів, та застосування моделей машинного навчання для визначення потенційної взаємодії користувачів з новими елементами. Застосування машинного навчання може включати прогнозування інтересів на основі подібних користувацьких профілів або попередніх тенденцій взаємодії.

У підсумку, розв'язання проблеми холодного старту вимагає глибокого розуміння як користувацької поведінки, так і технічних аспектів рекомендаційних систем. Поєднання різних підходів та постійне оновлення моделей на основі нових даних є критично важливим для підтримки актуальності та ефективності цих систем.

1.4 Розгляд існуючих способів вирішення проблеми

Фільтрування на основі контенту. Цей метод аналізує характеристики елементів та порівнює їх із інтересами користувача, визначеними через його попередні взаємодії або явно виражені переваги. Переваги цього підходу включають його здатність рекомендувати елементи, які є новими або нещодавно доданими до системи, оскільки не потрібні дані про попередні взаємодії з іншими користувачами. Однак, цей метод може бути обмежений тим, що він не враховує комплексні поведінкові шаблони користувачів і може пропонувати рекомендації, які не повністю відображають їхній потенціал до відкриття нового.

Модель на основі популярності. Цей підхід рекомендує елементи, які є популярними серед широкої аудиторії, і часто використовується як

простий спосіб запровадження рекомендацій для нових користувачів. Недоліком є те, що популярність не завжди еквівалентна персональним інтересам користувача, що може призвести до нерелевантних рекомендацій.

Гібридні моделі. Гібридні системи поєднують контент-орієнтоване фільтрування та колаборативну фільтрацію, намагаючись зібрати переваги обох підходів. Вони можуть використовувати контент для ініціації рекомендацій, а потім доповнювати та розвивати ці рекомендації на основі поведінкових даних користувачів, як тільки вони стають доступними. Це дозволяє системі бути більш гнучкою та ефективною при рекомендації нових продуктів та адаптації до нових користувачів [8].

Контекстно-залежні рекомендації. Ці системи враховують контекст, в якому елементи будуть використовуватись користувачем, включаючи час доби, розташування, та інші зовнішні фактори, які можуть впливати на вибір користувача. Такий підхід може допомогти вирішити проблему холодного старту, надаючи вагу актуальним і релевантним рекомендаціям [9].

Соціальне фільтрування. Соціальні фільтри включають у рекомендаційний процес соціальні зв'язки користувачів, їхні вподобання та взаємодії з іншими в соціальних мережах. Це може бути особливо корисно для нових користувачів, оскільки соціальні рекомендації можуть допомогти в швидкому набутті інформації про їхні переваги.

Методи обробки природної мови. NLP може бути використане для вдосконалення контент-орієнтованих систем, аналізуючи описи продуктів та відгуки користувачів для кращого розуміння семантики та контексту, що лежить в основі користувацьких переваг. Розвиток методів, таких як сентимент-аналіз, тематичне моделювання та семантичне аналізування, дозволяє виявляти неявні інтереси та переваги, забезпечуючи більш точні рекомендації для нових користувачів [10].

Використання метаданих і зовнішніх джерел. Інтеграція метаданих з зовнішніх джерел, таких як соціальні медіа та інші бази даних, може забезпечити додатковий контекст для рекомендацій, що дозволяє

рекомендаційним системам виходити за рамки обмеженої взаємодії всередині платформи.

Всі ці методи сприяють вирішенню проблеми холодного старту, але важливо розуміти, що не існує універсального рішення. Кожен підхід має свої умови ефективності, які повинні враховуватися при розробці та впровадженні рекомендаційних систем. Вибір оптимального методу часто залежить від специфіки домену, наявності даних, та індивідуальних вимог користувачів.

1.4.1 Загальні підходи надання рекомендацій

Асоціаційні правила є одним з основних методів у системах рекомендацій, зокрема в роздрібній торгівлі та електронній комерції. Цей метод аналізує шаблони покупок або вибору користувачів, щоб виявити, які товари часто купуються разом. Асоціаційні правила генерують рекомендації, засновані на зв'язках типу «якщо-то» (if-then), які вказують на ймовірність покупки певних товарів разом. Наприклад, якщо користувач купує каву, то ймовірно, що він також купить молоко.

Бізнес-правила відіграють критичну роль у процесі рекомендацій, оскільки дозволяють врахувати стратегічні цілі та обмеження компанії. Вони можуть включати логіку для просування нових продуктів, маржинальних товарів або товарів, що мають високий прибуток. Внесення бізнес-правил допомагає забезпечити, що рекомендації відповідають не тільки інтересам користувачів, а й стратегічним напрямам підприємства.

Бізнес правила є важливими, якщо ми хочемо скоригувати поведінку асоціаційних правил. Наприклад, існують люди, які разом дивляться і мультфільми, і хоррори. Якщо ми почнемо рекомендувати хоррори всім іншим людям, які переглядають мультфільми, це навряд чи буде влучною рекомендацією. У такому випадку ми можемо встановити бізнес-правило, яке забороняє таку рекомендацію, навіть якщо деякий процент користувачів

так і робить.

Можна використовувати безпосередній збір інформації від користувачів щодо їхніх переваг через інтерактивний інтерфейс. Це може бути реалізовано у формі опитування при першому відвідуванні платформи, де користувачів просять вказати категорії продуктів, жанри музики чи книг, що їх цікавлять. Цей підхід дозволяє системі негайно отримати важливу інформацію, яка може бути використана для ініціалізації персоналізованих рекомендацій.

Обов'язкове запитування може покращити персоналізацію, але також може стати перешкодою для користувачів, які бажають швидко почати користування сервісом. Важливо знайти баланс між збором даних та ненав'язливістю. Також інтереси користувача можуть змінюватися, тому необхідно постійно адаптувати рекомендації на основі нових даних.

1.5 Оцінки та метрики подібності

1.5.1 Оцінки

Для повноти дослідження важливо включити аналіз типів оцінок, які користувачі залишають у рекомендаційних системах, а саме розрізняючи явні (explicit) та неявні (implicit) оцінки. Це дозволить глибше зрозуміти, як користувачі взаємодіють із системою та які дані можуть бути використані для покращення точності рекомендацій.

Явні оцінки – це прямі вказівки користувачів про їхнє ставлення до елементів, таких як фільми, книги чи продукти. Це можуть бути оцінки за шкалою (наприклад, від 1 до 5 зірок), лайки або нелайки, або ж текстові відгуки. Явні оцінки забезпечують чітке уявлення про переваги користувачів і вважаються надійним джерелом для тренування рекомендаційних систем, оскільки безпосередньо відображають користувацькі уподобання.

Неявні оцінки визначаються на основі поведінки користувачів, що не передбачає прямого вираження їхніх переваг. Це можуть бути дані про перегляди сторінок, час, проведений на деякому контенті, історія покупок, частота та послідовність відвідувань, і таке інше. Хоча такі дані можуть бути менш конкретними порівняно з явними оцінками, вони важливі для визначення «тихих» переваг користувачів, особливо коли явні вираження відсутні.

Оцінки користувачів, як явні, так і неявні, мають бути інтегровані у модель рекомендаційної системи для забезпечення більш точної та персоналізованої взаємодії [11]. Аналіз цих оцінок допомагає системі краще розуміти користувацькі інтереси та настроювати рекомендації відповідно до зібраних даних. Таким чином, важливо розробити механізми для ефективної обробки та аналізу обох типів оцінок у контексті багатомірного підходу до аналізу користувацької поведінки.

Неявні оцінки мають особливе значення при вирішенні проблеми холодного старту в рекомендаційних системах. Неявні оцінки можна збирати автоматично з перших моментів взаємодії користувача з системою. Дані, такі як перегляди сторінок, час, проведений на певних статтях, або повторні візити, можуть швидко накопичуватися без необхідності активної участі користувача. Також неявні дані допомагають виявити скриті інтереси користувачів, які можуть не бути ясними через явні оцінки. Наприклад, користувач, який читає багато статей на певну тему, навіть якщо не ставить їм високих оцінок, демонструє зацікавленість у цій темі.

Для того, щоб збирати неявні оцінки і влучно їх використовувати, необхідно добре розуміти домен рекомендаційної системи. Домен визначає контекст, специфіку контенту та користувацькі взаємодії, що мають вирішальний вплив на способи обробки даних та вибір алгоритмів.

1.5.2 Метрики подібності

Для того, щоб зрозуміти, наскільки подібні 2 документа, необхідно обрати метрику подібності. В контексті обробки природної мови (NLP) і аналізу текстів, метрики вимірювання схожості використовуються для порівняння семантичної або синтаксичної подібності між словами, фразами або документами. Ці метрики допомагають в різних завданнях, від пошуку схожих документів до кластеризації текстів [12].

Cosine Similarity–це метрика, яка вимірює косинус кута між векторами у векторному просторі. Вона широко використовується для вимірювання схожості між векторними представленнями слів або текстів. Висока схожість означає, що об'єкти знаходяться близько один до одного у просторі.

Формула для розрахунку косинусної схожості може бути виражена як:

$$AdjustedCosine(i,j) = \frac{\sum_{u \in U_i \cap U_j} s_{ui} \cdot s_{uj}}{\sqrt{\sum_{u \in U_i \cap U_j} s_{ui}^2} \cdot \sqrt{\sum_{u \in U_i \cap U_j} s_{uj}^2}}, \quad (1.1)$$

де U_i – множина індексів користувачів, які оцінили предмет i;

U_j – множина індексів користувачів, які оцінили предмет j;

s_{ui} – середньоцентричний рейтинг користувача u для предмета I;

s_{uj} – середньоцентричний рейтинг користувача u для предмета j.

Ця подібність називається скоригованою косинусною подібністю, оскільки оцінки усередніли перед обчисленням значення подібності.

Подібність на основі кореляції–це метод оцінювання схожості між двома об'єктами на основі їх числових характеристик або ознак. В контексті обробки текстів, цей метод може бути застосований для оцінювання схожості між векторами слів або іншими числовими представленнями текстових даних.

Основним поняттям в подібності на основі кореляції є коефіцієнт

кореляції, який вимірює ступінь лінійної залежності між двома величинами. Чим ближче значення коефіцієнта кореляції до 1 або -1, тим сильніша залежність між величинами. Зазвичай використовується коефіцієнт кореляції Пірсона, але іноді можуть застосовуватися інші метрики кореляції, такі як Спірмена чи Кендалла.

$$r = \frac{\sum_{u \in U_i \cap U_j} (r_{iu} - \mu_i) \cdot (r_{ju} - j)}{\sqrt{\sum_{u \in U_i \cap U_j} (r_{iu} - \mu_i)^2} \cdot \sqrt{\sum_{u \in U_i \cap U_j} (r_{ju} - \mu_j)^2} \cdot}, \quad (1.2)$$

де U_i – множина індексів користувачів, які оцінили предмет i;

U_j – множина індексів користувачів, які оцінили предмет j;

r_{iu} – оцінка предмета i користувачем u;

r_{ju} – оцінка предмета j користувачем u;

μ_i – середній рейтинг предмета I;

μ_j – середній рейтинг предмета j.

Загалом, частіше використовують саме подібність на основі косинусів, порівняно з коефіцієнтом кореляції Пірсона.

Надійність функції часто залежить від кількості спільних рейтингів між двома користувачами. Якщо ця кількість невелика, необхідно застосувати коефіцієнт зниження для зменшення важливості такої пари користувачів. Знижена функція подібності виражається наступним чином:

$$DiscountedSim\ (u, v) = Sim(u, v) \cdot \frac{\min\{|I_u \cap I_v|, \beta\}}{\beta}, \quad (1.3)$$

де β – обраний поріг;

I_u – множина індексів предметів, оцінених користувачем u;

I_v – множина індексів предметів, оцінених користувачем v.

Тепер розглянемо, як визначати рейтинг цільового предмета t для користувача u. Перший крок – визначити топ-k найбільш подібних

елементів на основі обраної метрики подібності. Середнє зважене значення цих (вихідних) рейтингів подається як передбачене значення. Вага елемента j у цьому середньому розраховується як скоригована косинусна подібність між елементом j та цільовим елементом t.

Таким чином, передбачена оцінка $\widehat{r_{ut}}$ користувача u для цільового елемента t визначається наступним чином:

$$\widehat{r_{ut}} = \frac{\sum_{j \in Q_t(u)} AdjustedCosine(j,t) \cdot r_{uj}}{\sum_{j \in Q_t(u)} |AdjustedCosine(j,t)|}, \quad (1.4)$$

де $Q_t(u)$ – топ-k схожих елементів до елемента t, оцінених користувачем u;

r_{uj} – оцінка предмета j користувачем u.

Основна ідея полягає в тому, щоб використовувати власні оцінки користувача на подібні елементи на останньому етапі прогнозування. Наприклад, у системі рекомендацій фільмів група рівних елементів зазвичай буде складатися з фільмів подібного жанру. Історія оцінок того самого користувача на такі фільми є дуже надійним показником інтересів цього користувача.

1.6 Етичні аспекти та приватність даних

Рекомендаційні системи збирають та аналізують великі обсяги даних про користувачів, включаючи історію переглядів, покупок та взаємодій з контентом. Ця інформація є надзвичайно цінною, оскільки дозволяє системам точно ідентифікувати переваги та інтереси користувачів. Проте, це також породжує питання щодо того, наскільки етично використовувати особисті дані без чіткого розуміння того, як це впливає на приватне життя особи [13]. Важливим аспектом є забезпечення згоди користувачів на збір їхніх даних, яка має бути інформованою та чіткою, щоб користувачі розуміли, як їхні дані будуть використані.

Однією з ключових проблем у рекомендаційних системах є відсутність прозорості алгоритмів. Користувачі часто не розуміють, на якій основі система вирішила рекомендувати певний контент. Це може призвести до недовіри, оскільки користувачі можуть сприймати рекомендації як маніпулятивні або не відповідні їхнім реальним інтересам. Забезпечення прозорості в тому, як рекомендації генеруються, є критично важливим для підтримки довіри користувачів.

Анонімізація даних є одним із способів зниження ризиків, пов'язаних з приватністю користувачів. Це включає техніки видалення або заміни ідентифікуючої інформації в даних користувачів, щоб унеможливити їх пряме або непряме ідентифікування [14]. Використання анонімізованих даних може допомогти знизити правові та етичні ризики, забезпечуючи водночас високий рівень персоналізації.

1.7 Предметні області використання NLP для вирішення проблеми холодного старту

Застосування методів обробки природної мови (NLP) для вирішення проблеми холодного старту в рекомендаційних системах може мати різну ефективність залежно від конкретної предметної області. В деяких випадках, як на сайтах з науковими статтями, текстовий контент є плідним грунтом для NLP, оскільки він забезпечує багату інформацію для аналізу та витягування знань. У контексті соціальних мереж, хоча текстового контенту може бути менше, аналіз коментарів і описів може виявити корисну інформацію про соціальні зв'язки та переваги користувачів. У цьому випадку, NLP може виступати як допоміжний інструмент для підвищення точності рекомендацій.

Кожен домен має свої унікальні характеристики та вимоги до контенту. Наприклад, рекомендаційна система для фільмів повинна враховувати жанри, акторів, режисерів та інші метадані, тоді як система для

наукових публікацій зосереджується на ключових словах, авторах та цитуваннях. Розуміння цих аспектів допомагає точніше моделювати взаємодії між користувачами та об'єктами.

Різні домени мають різні типи користувачів із різними потребами та поведінкою. Наприклад, в електронній комерції користувачі можуть шукати конкретні продукти на основі ціни або якості, тоді як у музичних сервісах вони можуть шукати настрій або жанр. Розуміння цих відмінностей дозволяє розробити рекомендаційні системи, які краще задовольняють потреби конкретних користувачів.

Знання контексту домену допомагає системі рекомендацій створювати більш персоналізовані та релевантні пропозиції. Наприклад, в академічних рекомендаційних системах важливо знати актуальність певних дослідницьких тем або популярність конференцій.

Домен специфіка може впливати на вибір алгоритмів для рекомендацій. Наприклад, для доменів, де актуальність контенту швидко змінюється, можуть бути кращими алгоритми, які швидко адаптуються до нових трендів. В інших випадках можуть бути більш підходящими методи, що зосереджуються на довгостроковій історії користувацької взаємодії.

В окремих доменах можуть діяти специфічні законодавчі або етичні обмеження щодо збору та використання даних. Наприклад, у медичних та фінансових рекомендаційних системах високі вимоги до конфіденційності та безпеки даних.

Ось декілька областей, де методи обробки природної мови можуть бути корисними:

– електронна комерція – аналіз відгуків користувачів, описів товарів, і профілів покупців може забезпечити глибоке розуміння індивідуальних переваг та поведінки споживачів, сприяючи кращій персоналізації пропозицій;

– освіта – автоматизація аналізу навчальних матеріалів, наукових статей, і курсів може допомогти у визначенні релевантності

контенту для студентів та дослідників на основі їхніх академічних інтересів;

– фінанси – ефективний аналіз ринкових звітів, інвестиційних досліджень та фінансових новин може забезпечити інвесторам персоналізовану інформацію, що підвищує їхню здатність до прийняття обґрунтованих рішень;

– нерухомість – аналіз описів властивостей та користувацьких запитів може допомогти в адаптації пропозицій до персональних потреб покупців або орендарів, покращуючи шанси на успішні угоди;

– соціальні медіа – аналіз та інтерпретація користувацьких дописів, коментарів, та інших форм зворотного зв'язку, надаючи можливість зрозуміти тонкощі міжособистісних взаємин, інтересів та сентиментів, які можуть використовуватися для поліпшення соціальних зв'язків та вмісту, який користувачі отримують;

– культурна сфера та мистецтво – класифікація та рекомендації творів мистецтва, книг, фільмів, музичних творів та інших культурних подій, розкриваючи при цьому попередньо неявні зв'язки між різноманітними формами творчості та культурними перевагами користувачів.

У всіх цих областях використання NLP для рекомендаційних систем може суттєво поліпшити не тільки кількість, але й якість взаємодії між користувачем та платформою. Розуміння та обробка природної мови забезпечує важливу інформацію, яка може бути використана для зниження впливу проблеми холодного старту, особливо коли зібрані дані є обмеженими або не повністю відображають інтереси користувача. У комплексі з іншими даними, NLP забезпечує більш глибокий інсайт в переваги та поведінку користувачів, тим самим підвищуючи точність та релевантність рекомендацій, що є надзвичайно важливим для успішної роботи сучасних рекомендаційних систем.

1.8 Постановка задачі

Після детального аналізу існуючих рекомендаційних систем, їх моделей та способів вирішення проблеми холодного старту, ми переходимо до чіткого формулювання задачі, яка буде вирішуватися в рамках даної роботи. Мета цього підрозділу полягає в конкретизації завдань, які мають бути вирішені за допомогою розроблюваної рекомендаційної системи, і визначенні критеріїв їхнього виконання.

Основна задача, яку ми ставимо перед собою, – розробка та імплементація алгоритму рекомендаційної системи, який здатний ефективно вирішувати проблему холодного старту з використанням методів обробки природної мови.

Для досягнення поставленої мети у вирішенні проблеми холодного старту в рекомендаційних системах необхідно опрацювати наступні питання:

– сформувати та описати архітектуру рекомендаційної системи;
– обґрунтувати обрані методи та механізми;
– програмно реалізувати та навчити отриману модель;
– протестувати отриману модель та проаналізувати результати;
– окреслити можливі покращення та напрямки розвитку.

1.9 Висновки до розділу

У цьому розділі був проведений глибокий аналіз рекомендаційних систем. Були вивчені різні моделі рекомендацій, які включають колаборативну фільтрацію, контент-базову фільтрацію та гібридні системи. Також було розглянуто проблему холодного старту і існуючі методи її вирішення.

Приділено увагу етичним аспектам та забезпеченню приватності даних, що є необхідним для застосування майбутнього алгоритму у

реальних системах. Також розглянуто метрики подібності та порівняно, які з них будуть працювати у наших умовах.

Один з підрозділів присвячено розгляду предметних областей, у яких можливо вирішити проблему холодного старту з використанням NLP.

Розділ завершується постановкою задачі для подальшого дослідження.

Цей розділ надав фундамент для дослідження, підготувавши ґрунт для детального аналізу та розробки алгоритмів у наступних розділах, з особливим фокусом на інтеграції NLP для оптисмізації рекомендаційних систем.

2 ВИБІР ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ ТА РОЗРОБКА АЛГОРИТМУ

2.1 Розгляд методів NLP

У цьому підрозділі ми детально розглянемо ключові методи NLP, які використовуються для трансформації текстових даних у формат, зручний для обробки алгоритмами машинного навчання. Ми також оцінимо, як кожен з цих методів може бути застосований для покращення роботи рекомендаційних систем, особливо в контексті подолання викликів, пов'язаних з холодним стартом.

Аналіз методів NLP в цьому підрозділі має на меті не лише надати глибоке розуміння доступних інструментів та їхнього потенціалу у контексті рекомендаційних систем, але й допомогти у виборі найефективніших підходів для реалізації нашого алгоритму. Цей аналіз стане фундаментом для подальшого вибору засобів вирішення задачі та розробки алгоритму, який буде розглянуто у наступних підрозділах.

Основні напрямки обробки природньої мови представлені далі.

Тематичне моделювання дозволяє автоматично виявляти теми в текстових даних. Це допомагає розуміти інтереси нових користувачів та контенту, навіть без їхньої історії взаємодії. Наприклад, на основі текстових описів товарів та інформації з профілів користувачів можна виявити основні теми, які цікавлять нового користувача і рекомендувати товари, що відповідають цим темам [15].

Сентимент-аналіз допомагає визначити емоційний тон тексту, такий як позитивний, негативний або нейтральний [16]. Це може бути використано для аналізу відгуків нових користувачів або інших текстових даних, щоб зрозуміти їхні вподобання та реакції на різні продукти чи послуги.

Методи класифікації тексту дозволяють автоматично призначати

категорії до текстів. Вони можуть бути використані для категоризації контенту (наприклад, товарів, статей, відео) за тематикою, жанром або іншими параметрами. Методи класифікації текстів можуть бути побудовані таким чином, щоб адаптивно змінювати категорії на основі зміни інтересів користувача або властивостей контенту. Це дозволяє системі швидше реагувати на нові вхідні дані та надавати більш точні та персоналізовані рекомендації. Класифікація текстів також може бути використана для покращення процесу ранжування рекомендацій [17]. Наприклад, можна використовувати класифікацію тексту для ідентифікації контенту, який найбільше відповідає інтересам нового користувача, та надавати йому більш вагому впливу у процесі ранжування.

Аналіз ключових слів та фраз допомагає визначити важливі теми та концепції у текстах. Наприклад, шляхом аналізу текстової інформації, що надається при реєстрації, система може виявити ключові слова та теми, які цікавлять користувача, і використовувати цю інформацію для надання персоналізованих рекомендацій. Цей метод можна використати у поєднані з попереднім, класифікувавши контент за знайденими ключовими словами [18].

Векторні представлення слів дозволяють розуміти семантику слів та їх контекстуальні зв'язки. За допомогою векторних представлень слів можна виміряти семантичну схожість між словами та фразами, та застосувати один із попередніх методів – семантичний аналіз. На основі векторних представлень слів система може створювати персоналізовані вектори інтересів для кожного користувача. Векторні представлення слів є ефективним і масштабованим підходом до аналізу текстових даних. Вони дозволяють швидко обробляти та аналізувати великі обсяги тексту, що робить їх ідеальними для застосування у рекомендаційних системах з великою кількістю користувачів та контенту.

Глибоке навчання дозволяє створювати складні моделі для аналізу текстів та генерації контенту. Глибоке навчання може бути використане для

генерації контенту, який відповідає інтересам та вподобанням користувачів. Наприклад, система може використовувати глибокі моделі для створення персоналізованих описів товарів або статей. Глибокі нейронні мережі можуть бути використані для моделювання поведінки користувачів та прогнозування їхніх дій. Це дозволяє системі надавати рекомендації, які найбільш ймовірно зацікавлять нових користувачів.

2.2 Крос-модальне навчання в системах NLP

Крос-модальне навчання відіграє ключову роль у покращенні взаємодії між різними видами даних, що дозволяє системам глибше розуміти контекст і зміст, втілюючи це у вигляді більш точних рекомендацій та аналітики. Цей пункт розглядає основні техніки та методи крос-модального навчання, які інтегрують текстову, візуальну, і аудіо інформацію, створюючи синергію між ними.

Один із основних підходів у крос-модальному навчанні – створення спільного векторного простору для тексту та зображень. Такі моделі, як CLIP від OpenAI, навчаються розпізнавати візуальний контент з текстовими описами, використовуючи велику кількість образів та відповідних їм описів для створення загальних вбудовувань. Цей метод демонструє високу ефективність у зв'язуванні текстових та візуальних данних, що особливо корисно для рекомендаційних систем, де опис продукту та його візуальне представлення мають відповідати один одному.

Трансформери, такі як BERT (Bidirectional Encoder Representations from Transformers) та GPT (Generative Pre-trained Transformer), спочатку були розроблені для обробки тексту, але їх архітектура легко адаптується до інших видів даних. Наприклад, Vision Transformer (ViT) використовує подібні архітектурні принципи для обробки зображень, розбиваючи їх на дрібні сегменти і обробляючи як послідовність. Інтеграція ViT з BERT може дозволити системі одночасно аналізувати текст та зображення, підвищуючи

точність визначення контексту та смислу як в текстах, так і в візуальних матеріалах.

Механізм уваги в нейронних мережах, який був популяризований моделями трансформерів, можна також використовувати для крос-модального навчання. Цей підхід дозволяє моделі фокусуватися на ключових елементах у різних типах даних, наприклад, виділяючи важливі слова в тексті, які стосуються певних об'єктів на зображеннях або важливі моменти в аудіо треках. Така техніка може значно покращити здатність системи інтегрувати інформацію з різних джерел та надавати більш точні рекомендації [19].

Генеративні змагальні мережі (GANs) є ще одним прогресивним напрямком у крос-модальному навчанні. Вони використовуються для генерації нових даних, що відповідають реальним, застосовуючи дві мережі: генеративну, яка створює дані, та дискримінативну, яка визначає, чи є дані справжніми. В контексті крос-модального навчання, GANs можуть бути використані для створення синхронізованих текстових описів для зображень або навпаки. Цей підхід є особливо корисним для поліпшення якості зв'язку між різними типами контенту, наприклад, у створенні більш точних і природніх описів для візуального контенту на вебсайтах електронної комерції.

Контрастивне навчання – це техніка, яка використовується для вдосконалення здатності моделей виявляти відмінності та схожості між різними даними, що мають бути порівняні безпосередньо або ж порівняно з великою групою альтернатив. У крос-модальному контексті, цей метод може допомогти підвищити ефективність моделей, навчених співвідносити текст до відповідних зображень або відео, виокремлюючи ключові особливості кожного медіа. Завдяки контрастивному навчанню, системи можуть краще «розуміти» зміст зображень та асоціювати їх з відповідними текстовими описами, що важливо для автоматизованих систем модерування контенту або розширених рекомендаційних систем.

Останнім часом важливість інтеграції метаданих і контекстуальної інформації стає все більшою. Ці дані можуть включати час, місце зйомки фотографії, інформацію про автора тексту чи відео, тощо. Використання цієї інформації може значно підвищити точність крос-модальних систем, оскільки моделі навчаються враховувати не лише безпосередні дані, але й контекст, в якому вони використовуються. Наприклад, система може використовувати час та геолокацію зображення для покращення точності визначення його контексту та можливого змісту.

2.3 Вибір сфери та напрямку NLP

У цій роботі ми сфокусуємось на вирішенні проблеми холодного старту для нового об’єкту. У якості предметної області я обрала технічні статті. Технічні статті часто створюються та публікуються великою кількістю авторів та видавництв, що призводить до великого обсягу доступних даних. Це дозволяє побудувати масштабні моделі та провести розгорнуті дослідження.

Ця сфера постійно розвивається та оновлюється з новими відкриттями та технологічними проривами. Це створює постійний потік нової інформації, яка може бути використана для аналізу та надання актуальних рекомендацій. В області технічних статей часто існує активна спільнота фахівців, які обговорюють та обмінюються інформацією. Це може бути використано у сентиментальному аналізі.

Робота з технічними статтями може вимагати вдосконалення методів NLP для обробки технічної термінології, складних формулювань та великої кількості скорочень та акронімів. Це відкриває можливості для розвитку нових методів та підходів у сфері обробки тексту.

З описаних вище методів, я зосереджусь на використанні векторних представлень слів для виміру схожості між текстами, а також застосую модель класифікування тексту для поділу на теми, жанри, рівні складності.

Це необхідно для того, щоб порівнювати не два тексти на абсолютно різні теми (які, скоріш за все, матимуть низький рівень схожості), а щоб робити це між текстами з більшою вірогідністю бути схожими.

2.4 Семантичні методи

Семантичні методи в NLP відіграють критичну роль у розумінні та обробці мовних структур на більш глибокому, семантичному рівні. Ці методи спрямовані на вдосконалення здатності машин розуміти справжнє значення текстів, що є ключовим для багатьох застосувань, включаючи машинний переклад, автоматичне анотування текстів та рекомендаційні системи.

Латентний семантичний аналіз є одним із старіших, але все ще вкрай ефективних методів для виявлення зв'язків між словами та контекстами в великих текстових корпусах. Через використання технік розкладу матриць, таких як сингулярний розклад (SVD), LSA дозволяє виявити семантичні структури в даних, зводячи слова та текстові документи до низьковимірного семантичного простору. Це допомагає у вирішенні проблеми синонімії та полісемії, оскільки семантично близькі слова мапуються близько одне до одного.

LDA є популярним методом тематичного моделювання, що дозволяє ідентифікувати теми, які переважають у колекціях текстових документів. Цей метод базується на ймовірності та дозволяє кожному документу бути представленим як суміш тем, де кожна тема характеризується розподілом слів. LDA особливо корисний для організації великих текстових даних, аналізу вмісту та дослідженні зміни тем в часі.

WordNet – це багатомовна лексична база даних, де слова згруповані за значеннями і пов'язані різними семантичними зв'язками, такими як синонімія, антонімія, іпонімія. Використання WordNet у NLP може значно покращити здатність алгоритмів до розуміння семантичних відносин між

словами, що є важливим для задач, таких як семантичний пошук, вирішення проблеми багатозначності слів, і автоматичне генерування тексту.

Семантичні методи можуть бути застосовані для поліпшення точності та релевантності рекомендацій у системах, які використовують текстові дані. Інтегрування семантичного аналізу дозволяє системам краще розуміти зміст користувацьких відгуків, описів продуктів, і новинних статей, сприяючи більш точному виявленню інтересів та переваг користувачів.

2.5 Методи векторного представлення

2.5.1 Модель «Мішок слів»

Модель «Мішок слів» є одним з простих та ефективних методів для представлення тексту у векторній формі у сфері обробки природної мови (NLP). Цей підхід дозволяє аналізувати та розуміти текстові дані, ігноруючи порядок слів у реченні та враховуючи лише наявність слів у документі.

Ось як виглядають основні концепці та кроки цієї моделі. Перший крок у моделі «Мішок слів»–це розділення тексту на окремі слова або токени. Цей процес відомий як токенізація. Після токенізації створюється словник, що містить всі унікальні слова, які зустрічаються у корпусі текстів. Кожне слово отримує унікальний індекс у цьому словнику. Кожен документ (або речення) у корпусі текстів представляється у векторній формі. Розмірність вектора відповідає кількості слів у словнику. Кожен елемент вектора відображає кількість входжень відповідного слова у документі.

Модель «Мішок слів» широко використовується в NLP для багатьох завдань, таких як класифікація тексту, кластеризація, аналіз настроїв тощо. Вона дозволяє легко враховувати велику кількість слів та текстів і може бути використана як базова модель для подальших досліджень і розробок в

області NLP.

2.5.2 Модель Word2Vec

Модель Word2Vec є потужним інструментом в обробці природної мови, призначеним для отримання векторних представлень слів з текстових даних. Цей підхід революціонізував обробку тексту, дозволяючи комп'ютерним моделям розуміти семантичні взаємозв'язки між словами та здійснювати подібність між ними.

Word2Vec використовує дві основні архітектури: Continuous Bag of Words (CBOW) та Skip-gram. CBOW намагається передбачити слово, виходячи з контексту, тоді як Skip-gram передбачає контекст, виходячи з слова. Обидва підходи навчаються на основі зв'язків між словами у великому корпусі тексту.

Модель Word2Vec базується на нейронних мережах, які навчаються векторизувати слова в просторі з невеликою кількістю вимірів. Вона використовується для передбачення контексту слова в реченні або, навпаки, слова на основі його контексту.

Одним з головних досягнень Word2Vec є отримання векторних представлень слів. Кожне слово у тексті представляється у векторному просторі, де подібні слова розташовані близько одне до одного. Ці векторні представлення можуть бути використані для різноманітних завдань NLP, таких як класифікація тексту, машинний переклад, аналіз настроїв та багато інших.

Перевагами цієї моделі є здатність до захоплення семантичних зв'язків між словами, ефективність та швидкість навчання на великих корпусах тексту [20]. Із недоліків можна виділити потребу у значних обчислювальних ресурсах та об’єму даних, а також можливості впливу рідкісних або сленгових слів.

2.5.3 GloVe

GloVe (Global Vectors for Word Representation) є моделлю для отримання векторних представлень слів, яка вирішує проблему представлення слів у векторній формі з урахуванням глобальних статистичних залежностей між ними. Цей підхід дозволяє отримати більш точні та збалансовані векторні представлення, що враховують не тільки локальні контекстуальні відносини, а й глобальні структурні зв'язки між словами у великих корпусах тексту.

GloVe базується на аналізі статистики співвходження слів у великих корпусах тексту. На основі цих статистичних даних будується матриця співвходження.

За допомогою матриці співвходження обчислюється функція втрат, яка оцінює відповідність між векторними представленнями слів та їхніми співвходженнями. Головною метою є мінімізація цієї функції втрат.

Після оптимізації функції втрат отримуються векторні представлення слів, які враховують глобальні статистичні зв'язки між ними у корпусі тексту. Ці вектори можна використовувати для різних завдань обробки природної мови.

Перевагами цього методу є урахування глобальних статистичних зв'язків між словами, здатність до отримання збалансованих та точних векторних представлень слів. Недоліки тут приблизно такі ж, як і у попереднього методу.

2.6 Методи кластеризації

2.6.1 K-means кластеризація

Метод k-means є одним з найбільш популярних і широко використовуваних методів кластеризації в аналізі текстових даних. Цей

алгоритм ефективно групує схожі текстові документи у визначену кількість кластерів, базуючись на їх семантичних або стилістичних характеристиках, що дозволяє виявляти тематичні або стилістичні зв'язки між ними.

Перед застосуванням методу k-means, текстові дані потрібно перетворити в векторне представлення. Це можна зробити за допомогою методів векторизації, таких як TF-IDF, який оцінює важливість слова у контексті документа та датасету, або використання моделей Word2Vec чи GloVe, які генерують більш глибокі семантичні представлення слів.

Важливим кроком у застосуванні k-means є визначення кількості кластерів. Це може бути виконано на основі експертних знань або за допомогою методів, які допомагають визначити оптимальну кількість кластерів, наприклад, метод ліктя, який аналізує варіацію відносно кількості кластерів і дозволяє вибрати кількість, при якій збільшення кластерів не призводить до значного покращення зміни варіації.

K-means працює шляхом визначення центроїдів для кожного кластера, які спочатку вибираються випадково, та призначення кожного документа до найближчого кластера на основі відстані між вектором документа і центроїдом. Алгоритм потім ітеративно оновлює центроїди, засновані на середніх значеннях векторів у кластері, до тих пір, поки не буде досягнуто збіжності або не буде виконано максимальну кількість ітерацій.

Метод k-means має ряд переваг, таких як простота реалізації, швидкість і масштабованість, які роблять його ефективним при роботі з великими обсягами даних. Однак метод має і недоліки: він чутливий до вибору початкових центроїдів та кількості кластерів, а також може показувати погані результати у випадках, коли дані мають нелінійні структури або присутні шуми та аутлаєри.

Ці характеристики роблять k-means універсальним інструментом для аналізу текстових даних, здатним ефективно виявляти складні тематичні зв'язки в документах, хоча і з вимогою до ретельного підходу у виборі параметрів для конкретного застосування.

2.6.2 Ієрархічна кластеризація

Ієрархічна кластеризація є іншим популярним методом групування текстових документів разом у кластери. Вона відрізняється від k-means тим, що не потребує передбачення кількості кластерів перед кластеризацією та дає змогу створювати деревоподібні структури кластерів.

У відмінність від k-means, де потрібно заздалегідь визначити кількість кластерів, ієрархічна кластеризація створює деревоподібну структуру кластерів, де кожен вузол представляє собою кластер, а гілки – підкластери [21].

Ієрархічна кластеризація може бути агломеративною (знизу вгору) або дивізійною (зверху донизу). У першому випадку алгоритм спочатку вважає кожен об'єкт окремим кластером, а потім поступово об'єднує їх у більші кластери. У другому випадку, навпаки, весь набір даних розглядається як один кластер, який поділяється на менші кластери з кожним подальшим кроком.

Для визначення схожості між документами використовуються різні метрики відстані, такі як евклідова, Манхеттенська, косинусна тощо.

Із переваг методу можна виділити відсутність потреби передбачати кількість кластерів. Алгоритм автоматично будує ієрархічну структуру кластерів, яка може бути відображена у вигляді дерева. Іншою перевагою є можливість аналізувати дані на різних рівнях деталізації. Це дозволяє користувачам отримувати інформацію як загального характеру, так і докладніше розглядати кластери на більш нижніх рівнях ієрархії. Також ієрархічна кластеризація використовує метрики відстані для визначення схожості між документами, що дозволяє робити об'єктивні рішення на основі семантичних або стилістичних характеристик.

Однією з основних недоліків ієрархічної кластеризації є висока обчислювальна складність, особливо для великих об'ємів даних. Побудова ієрархічної структури може бути витратною за часом та ресурсами. Як і в

інших методах кластеризації, ієрархічна кластеризація може бути чутливою до шуму та аутлаєрів у даних, що може призвести до формування неточних або недороблених кластерів. Іноді ієрархічна структура кластерів може бути складною для інтерпретації, особливо якщо вона має багато рівнів або складну структуру.

2.7 Розробка алгоритму

Для аналізу схожості технічних статей найбільш підходящим методом буде використання векторних представлень слів разом із моделями класифікації тексту, зокрема метод Word2Vec або GloVe для отримання векторних представлень слів та моделі кластеризації, яка базується на цих векторних представленнях.

Векторні представлення слів, отримані за допомогою Word2Vec або GloVe, дозволяють перетворити слова у вектори у просторі чисел, де семантично схожі слова розташовані близько одне до одного. Таким чином, схожість між двома технічними статтями може бути визначена шляхом порівняння векторних представлень їхніх слів. Статті, які мають схожі векторні представлення, можна вважати схожими за змістом.

Далі, для кластеризації технічних статей можна використовувати алгоритми кластеризації, такі як k-means або hierarchical clustering, які будуть базуватися на векторних представленнях слів. Ці алгоритми групують статті в кластери на основі схожості їхніх векторних представлень, тобто статті, які мають близькі векторні представлення, будуть призначені до одного кластера.

У процесі розробки цього алгоритму було виконано значну кількість експериментальних досліджень, які допомогли ідентифікувати оптимальні підходи до вирішення задачі холодного старту в рекомендаційних системах. Особливо корисним виявилось впровадження методів NLP для аналізу семантичної близькості між текстами, що значно підвищує релевантність і

точність рекомендацій. Ці результати підтверджують важливість подальшого дослідження в цьому напрямку та можливість його практичного застосування у різних сферах.

Методи Word2Vec і GloVe були обрані через їх здатність ефективно перетворювати слова в семантичні вектори, що відображають смислові зв’язки. Це забезпечує глибше розуміння тексту порівняно з традиційними методами, як «мішок слів». Кластеризація використовується як логічне продовження цього аналізу, дозволяючи групувати статті на основі схожості їхніх векторних представлень, що сприяє ефективному виявленню та організації інформації за темами.

Використання моделей Word2Vec і GloVe для генерації векторних представлень слів значно підвищує якість семантичного аналізу, оскільки ці моделі забезпечують глибоке розуміння контексту та семантичних відносин між словами. Це дозволяє не лише виділити схожість на основі повторення слова, але й зафіксувати більш глибокі семантичні зв'язки, що є критично важливим для точного аналізу текстів. Така можливість робить Word2Vec і GloVe особливо цінними для аналізу великих текстових корпусів, наприклад, технічних документів або наукових статей.

Крім того, масштабованість цих методів дозволяє обробляти великі набори даних без значної втрати швидкості або якості, що робить їх придатними для застосування в сучасних дослідницьких та комерційних проектах. Велика кількість інформації може бути ефективно синтезована і проаналізована, сприяючи розробці більш досконалих систем для рішення складних задач.

На завершення, гнучкість у виборі методів кластеризації, таких як k-means або hierarchical clustering, забезпечує можливість адаптувати аналіз під конкретні потреби проекту. Ці методи дозволяють групувати документи на основі схожості їхніх векторних представлень, що відкриває додаткові можливості для організації і сегментації даних. Таким чином, можна вирішувати специфічні задачі, пов'язані з класифікацією та аналізом

контенту, що є особливо корисним у сферах, де важливо точно зрозуміти великі обсяги текстової інформації.

Одним з недоліків використання Word2Vec і GloVe є їхня сильна залежність від контексту, в якому вживаються слова, що може призвести до помилок у визначенні схожості між документами, коли слова використовуються в різних контекстах. Також, алгоритми кластеризації можуть мати труднощі з визначенням оптимальної кількості кластерів або некоректним групуванням документів, особливо коли тематичні кластери перекриваються. Додатково, Word2Vec і GloVe можуть неефективно обробляти нові слова або жаргон, які не були представлені у тренувальних даних.

Ці недоліки можуть бути прийнятні залежно від контексту застосування алгоритму. Наприклад, у сценаріях, де база документів має відносно однорідний стиль написання та тематику, сильна залежність від контексту та проблеми з новими словами можуть мати менший вплив. Також, попри труднощі з кластеризацією, ці методи все ж забезпечують цінну інформацію для первинного групування документів, що може значно полегшити подальший аналіз.

Використання алгоритмів кластеризації разом із сучасними методами NLP створює нові можливості для підвищення ефективності рекомендаційних систем. Ці методи забезпечують не тільки швидкість та точність обробки даних, але й відкривають двері для глибшого аналізу зв'язків між користувацькими даними та контентом. Застосування цих підходів може значно поліпшити адаптацію рекомендаційних систем до потреб кожного користувача, що робить дані дослідження не тільки актуальними, але й перспективними для подальшого впровадження у широкий спектр застосувань.

2.8 Висновки до розділу

У другому розділі кваліфікаційної роботи було проведено аналіз та вибір засобів вирішення задачі з використанням методів обробки природної мови для оптимізації рекомендаційних систем. Значна увага була приділена детальному розгляду векторних представлень тексту, таких як модель «Мішок слів», Word2Vec та GloVe, які забезпечують основу для семантичного аналізу та збільшення точності рекомендацій. Вибір методів НЛП та їх застосування у сфері рекомендаційних систем визначилися через їхню здатність покращувати якість рекомендацій, особливо в контексті холодного старту, де традиційні підходи часто зазнають невдач.

Крім того, у цьому розділі було досліджено та застосовано різні методи кластеризації, включаючи K-means та ієрархічну кластеризацію, які дозволяють ефективно групувати великі обсяги даних та виділяти важливі шаблони в поведінці користувачів.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Вибір датасету

Для вибору датасету я буду враховувати такі критерії.

Важливо обрати датасет, який відображає тематику нашої роботи. Це дозволить нам робити більш змістовні висновки. Наприклад, можна вибрати датасет з технічними статтями про машинне навчання, нейронні мережі або обробку природних мов.

При виборі розміру важливо знайти золоту середину. З одного боку, великий обсяг даних дозволяє проводити більш комплексний та розгорнутий аналіз, а також отримувати більш точні результати. Датасет з великою кількістю статей також може представляти більшу різноманітність тем та стилів, що дозволяє зробити аналіз більш репрезентативним. З іншого боку, великий розмір вплине на час та складність виконання, а також отримані результати може бути складніше аналізувати.

Важливо обирати датасет з високоякісними технічними статтями, які містять інформацію, що відповідає вашим дослідницьким цілям. Зручний формат даних, наприклад, у вигляді текстових файлів або баз даних, спрощує їх обробку та підготовку для аналізу.

Також необхідно переконатися, що обраний датасет відповідає ліцензійним умовам та правовим обмеженням, які можуть виникнути під час використання даних. Датасет повинен бути доступний для використання та дослідження без обмежень.

Для своїх цілей я обрала датасет ArXiv. arXiv – це архів із відкритим доступом для майже 2,4 мільйонів наукових статей у галузях інформатики, фізики, математики, статистики, та багатьох інших тем. Він відповідає усім критеріям, які ми визначили для себе.

3.2 Опис датасету

Цей датасет є дзеркалом оригінальних даних ArXiv. Оскільки повний набір даних є досить великим (більше 1.1 терабайтів), цей датасет містить лише метадані у форматі JSON. Файл для кожної статті складається з наступної інформації:

– id: унікальний ідентифікатор ArXiv;
– submitter: хто подав статтю;
– authors: автори статті;
– title: назва статті;
– comments: додаткова інформація, така як кількість сторінок і фігур;
– journal-ref: інформація про журнал, в якому була опублікована стаття;
– doi: цифровий ідентифікатор об'єкта;
– abstract: абстракт статті;
– categories: категорії / теги у системі ArXiv;
– versions: історія версій.

Абстракт статті містить короткий опис її змісту і часто містить ключову інформацію, яка може бути корисною для аналізу та моделювання. Абстракт може містити основні теми, концепції, методи та результати дослідження, що дозволяє використовувати його для різноманітних завдань NLP. При використанні датасету arXiv, ми зможемо використовувати абстракти для отримання інсайтів щодо тематики та структури технічних статей.

Для наших цілей ми будемо використовувати саме абстракти для порівняння статей.

3.3 Аналіз та підготовка даних

Спочатку імпортуємо всі необхідні бібліотеки. Ми будемо використовувати pandas для роботи з даними, datetime для роботи з часлм, nltk для роботи з текстовими даними, spacy для обробки природної мови.

Тепер подивимось на розмір датасету (рисунок 3.1).

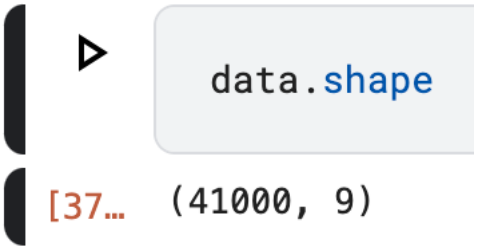

Рисунок 3.1 – Розмір датасету

Маємо 41 тисячу рядків записів. 9 стовпців-атрибутів та їх значення були перераховані у підрозділі вище.

Тепер виведемо перші 5 рядків датасету, щоб зрозуміти, як виглядають дані (рисунок 3.2). Можемо бачити, що стаття може мати декількох авторів та декілька тегів, а також належати до декількох категорій. Id складається з трьох частин – першого та другого числа, розділеними крапками, а також версії. Також наявні посилання на повну статтю на arXiv. Ми можемо цим скористатись при перевірці наданих рекомендацій, щоб впевнитись у схожості повного тексту та представних там інтересів.

Тепер необхіно перевірити повноту даних – чи наявні там пропуски, незаповнені поля, рядки, які сильно виділяються з-поміж остальних, щоб проводити подальший аналіз.

```
[4]: data.head()
```

[4]:

	author	day	id	link	month	summary	tag	title	year
0	[{'name': 'Ahmed Osman'}, {'name': 'Wojciech S...	1	1802.00209v1	[{'rel': 'alternate', 'href': 'http://arxiv.or...	2	We propose an architecture for VQA which utili...	[{'term': 'cs.AI', 'scheme': 'http://arxiv.org...	Dual Recurrent Attention Units for Visual Ques...	2018
1	[{'name': 'Ji Young Lee'}, {'name': 'Franck De...	12	1603.03827v1	[{'rel': 'alternate', 'href': 'http://arxiv.or...	3	Recent approaches based on artificial neural n...	[{'term': 'cs.CL', 'scheme': 'http://arxiv.org...	Sequential Short-Text Classification with Recu...	2016
2	[{'name': 'Iulian Vlad Serban'}, {'name': 'Tim...	2	1606.00776v2	[{'rel': 'alternate', 'href': 'http://arxiv.or...	6	We introduce the multiresolution recurrent neu...	[{'term': 'cs.CL', 'scheme': 'http://arxiv.org...	Multiresolution Recurrent Neural Networks: An ...	2016
3	[{'name': 'Sebastian Ruder'}, {'name': 'Joachi...	23	1705.08142v2	[{'rel': 'alternate', 'href': 'http://arxiv.or...	5	Multi-task learning is motivated by the observ...	[{'term': 'stat.ML', 'scheme': 'http://arxiv.o...	Learning what to share between loosely related...	2017
4	[{'name': 'Iulian V. Serban'}, {'name': 'Chinn...	7	1709.02349v2	[{'rel': 'alternate', 'href': 'http://arxiv.or...	9	We present MILABOT: a deep reinforcement learn...	[{'term': 'cs.CL', 'scheme': 'http://arxiv.org...	A Deep Reinforcement Learning Chatbot	2017

Рисунок 3.2 – Огляд датасету

Перевірка на нульові та пропущені значення показує (рисунок 3.3), що все добре, і датасет повністю заповнений.

```
data.isnull().sum()
```

```
[38… author     0
day        0
id         0
link       0
month      0
summary    0
tag        0
title      0
year       0
dtype: int64
```

Рисунок 3.3 – Нульові значення

Для подальшого аналізу ми будемо використовувати абстракти статей. Виділимо їх у окремий датафрейм (рисунок 3.4). Також переведемо у нижній регістр для полегшення майбутніх маніпуляцій.

Виведемо новий датафрейм та подивимось, як він виглядає.

```
[7]: data_abstract = data['summary'].str.lower()
     data_abstract.head()

[7]: 0    we propose an architecture for vqa which utili...
     1    recent approaches based on artificial neural n...
     2    we introduce the multiresolution recurrent neu...
     3    multi-task learning is motivated by the observ...
     4    we present milabot: a deep reinforcement learn...
     Name: summary, dtype: object
```

Рисунок 3.4 – Датасет абстрактів

Перейдемо до очищення даних. Застосуємо лематизацію – перетворення слова на його базову форму (наприклад, running замінюється на run). Також приберемо речення, де менше 2 слів. Робимо це через те, що обрана нами модель використовує контекстні слова для створення векторів. Якщо речення коротке, користь від такого навчання буде низькою.

Також видалимо всі символи, крім літер латинського алфавіту та апострофів, та переведемо речення у нижній регістр. Видалимо можливі дублікати, які створились після нашої обробки.

Перевіримо, наскільки зменшився розмір нашого датасету (рисунок 3.5).

```
[12]: data_clean = pd.DataFrame({'clean': txt})
      data_clean = data_clean.dropna().drop_duplicates()
      data_clean.shape

[12… (40955, 1)
```

Рисунок 3.5 – Розмір нового датасету

Прибралось всього 45 рядків.

3.4 Використання Word2Vec

Тепер використаємо бібліотеку Gensim для об’єднання багаторівневих сполучень в один термін. Phrases використовується для навчання моделі на основі фразового аналізу, а Phraser–для ефективного застосування навченої моделі.

Кожний рядок у стовпці «clean» датасету data_clean розділяється на окремі слова за допомогою методу split(). Результатом є список списків слів, де кожен внутрішній список містить слова одного рядка.

Перейдемо до створення важливих словосполучень

Біграми та триграми–це послідовності з двох або трьох слів в тексті, які з'являються разом з високою частотою і можуть мати спеціальне значення або виконувати певну функцію в мові. Вони є важливими для розуміння контексту та семантики тексту.

Для створення біграм, створюємо об’єкт Phrases. Мінімальну кількість зустрічань, яка потрібна для врахування словосполучення, у нашому випадку буде дорівнювати 30.

Тепер додамо біграми до перших слів, і створимо триграми. Кількість зустрічань, яка потрібна для врахування словосполучення, ставимо меншою – 20.

Вийшли отакі слова та словосполучення (рисунок 3.6).

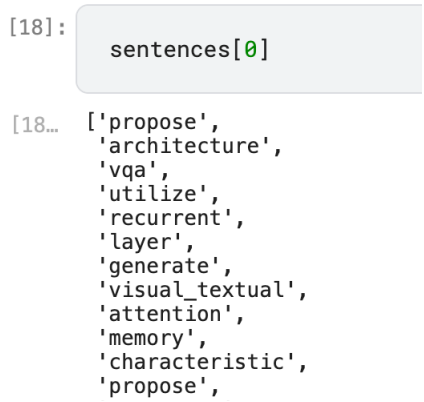

Рисунок 3.6 – Обрані слова

Порахуємо кількість унікальних слів у тексті та їхню частоту згадування.

Найвживаніші 10 слів та словосполучень з усього датасету (рисунок 3.7).

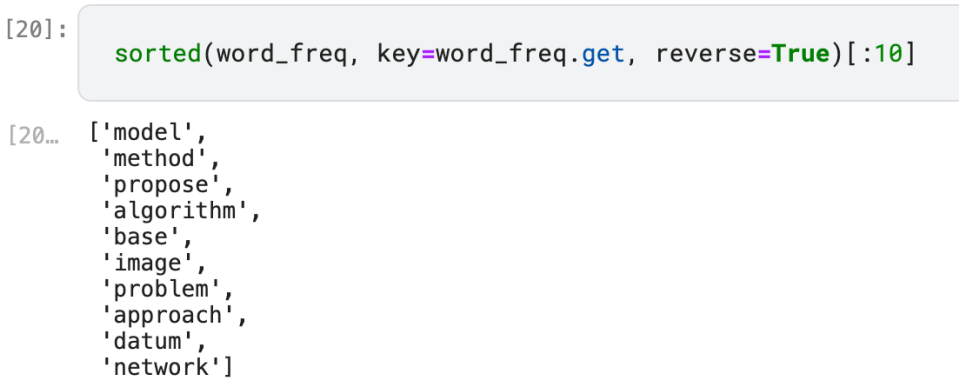

Рисунок 3.7 – Найчастіше вживані слова

Перейдемо до створення моделі Word2Vec для навчання векторних представлень слів. Розберемо, що означає кожен параметр, та які значення ми встановили:

– min_count: Цей параметр вказує мінімальну кількість згадувань слова у корпусі даних, необхідну для включення його до словникового корпусу. Слова, які зустрічаються рідше, ніж min_count (у нас – 15 разів), будуть проігноровані;

– window: Цей параметр вказує максимальну відстань між цільовим словом та словами контексту при навчанні моделі. У даному випадку, слова на відстані не більше 5 слів від цільового будуть враховані у контексті;

– size: Розмір векторів, що представляють кожне слово у векторному просторі. У даному випадку, вектори матимуть розмірність 300.

– sample: Цей параметр визначає поріг для випадкового відбору слів у частотному розподілі під час навчання. Слова, які частіше зустрічаються, можуть бути випадково замінені на інші слова. У даному випадку, поріг

встановлено на 6e-5;

– alpha та min_alpha: Параметри керують швидкістю зменшення шагу навчання під час тренування моделі. Початкове значення шагу навчання встановлено на 0.03, і зменшується до значення min_alpha з кожною епохою;

– negative: Цей параметр визначає кількість «негативних» зразків, які використовуються під час навчання моделі. «Негативні» зразки–це слова, які не належать до контексту цільового слова, і використовуються для навчання моделі для відмінності між справжніми словами та випадковими;

– workers: Цей параметр визначає кількість потоків, які використовуються під час навчання моделі. У даному випадку, використовується cores-1 потоків, де cores–кількість доступних ядер процесора.

Необхідний словник було створено приблизно за хвилину.

Запустимо тренування моделі (рисунок 3.8).

```
[25]: t = time()

      w2v_model.train(sentences, total_examples=w2v_model.corpus_count, epochs=25, # epochs=30,
                      report_delay=1)

      print('Time to train the model: {} mins'.format(round((time() - t) / 60, 2)))

Time to train the model: 10.83 mins
```

Рисунок 3.8 – Тренування моделі

Тепер у нас кожен абстракт представлений вектором слів. Таким чином, щоб отримати рекомендації для нової статті, необхідно створити вектор подібним шляхом, а після цього отримати схожі на неї.

Функція рекомендації виглядає наступним чином (рисунок 3.9). Спочатку ми виділяємо масив індексів. Далі рахуємо подібність і сортуємо за нею від найбільшого показника до найменшого. Після цього беремо 10

перших результатів і повертаємо їх.

```
[ ]:
indices = pd.Series(recommend_df.index, index=recommend_df['abstract']).drop_duplicates()

def get_recommendations(abstract, cosine_sim, indices):
    idx = indices[abstract]
    # Get the pairwsie similarity scores
    sim_scores = list(enumerate(cosine_sim[idx]))
    # Sort based on the similarity scores
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    # Get the scores for 10 most similar
    sim_scores = sim_scores[1:11]
    # Get the indices
    paper_indices = [i[0] for i in sim_scores]
    # Return the top 10 most similar
    return recommend_df['abstract'].iloc[paper_indices]
```

Рисунок 3.9 – Функція рекомендацій

3.5 Кластеризація

Але перед цим, додамо ще кластеризацію. Як зазначалося раніше, це допоможе шукати схожі статті лише у потрібному кластері. Це повинно зменшити час на пошук та трохи оптимізувати його (адже логічно, що схожі статті будуть належати до одного кластеру).

Як зазначалося раніше, будемо використовувати k-means. Спочатку необхідно обрати кількість кластерів. Для цього застосуємо метод ліктя (рисунок 3.10).

Основна ідея методу полягає у визначенні точки, де приріст варіативності всередині кластерів починає зменшуватися, додаванням більшої кількості кластерів, тобто моменту, коли збільшення кількості кластерів не призводить до суттєвого покращення якості кластеризації.

Для визначення точки необхідно застосувати метод на різній кількості кластерів. Ми обрали рамки від 1 до 80. У циклі встановлюємо кількість кластерів одному з чисел и дивимось на показник варіативності всередині кластера.

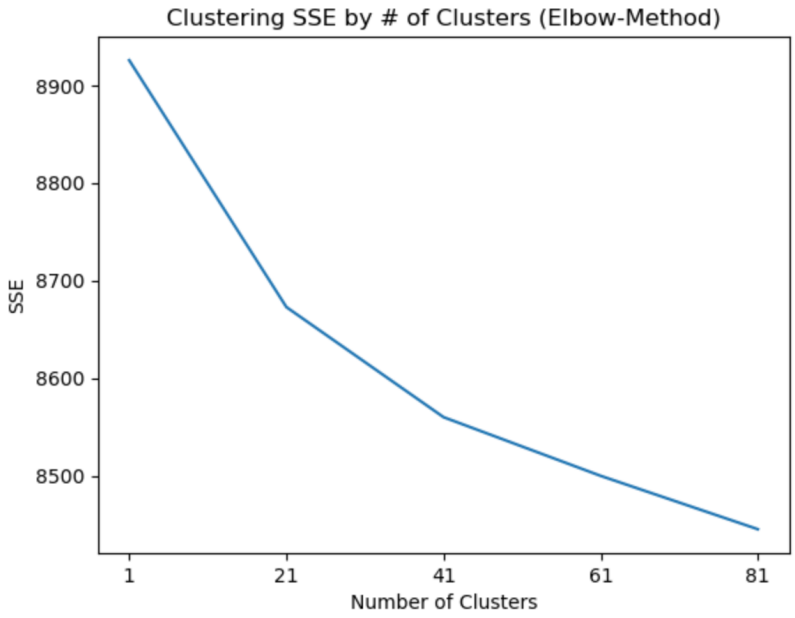

Рисунок 3.10 – Вибір розміру кластеру

Бачимо, що оптимальна кількість – близько 20 кластерів.

Після навчення моделі зробимо пару кроків для візуалізації нашої кластеризації. По-перше, створимо новий датасет, який складається із номеру кластеру, абстракту і назви статті. Після цього застосуємо T-SNE(T-Distributed Stochastic Neighbor Embedding) від SkLearn для генерації різноманіття матриці у нижчому вимірі. Цей підхід перетворює подібність точок у спільні ймовірності – мінімізуючи розбіжність ймовірностей об’єднання в даних високої розмірності та даних низької розмірності за допомогою градієнтного спуску.

Важливо пам’ятати, що створені візуалізації (рисунок 3.11) не є фактичною оцінкою простору ознак, вони є лише способом побачити поведінку даних у високих розмірах.

Більш низькі значення KL Divergence вказують на кращу відповідність між простором з великою розмірністю та представленням з низькою розмірністю.

Якщо навести курсор на точку, можна побачити додаткові деталі про

статтю.

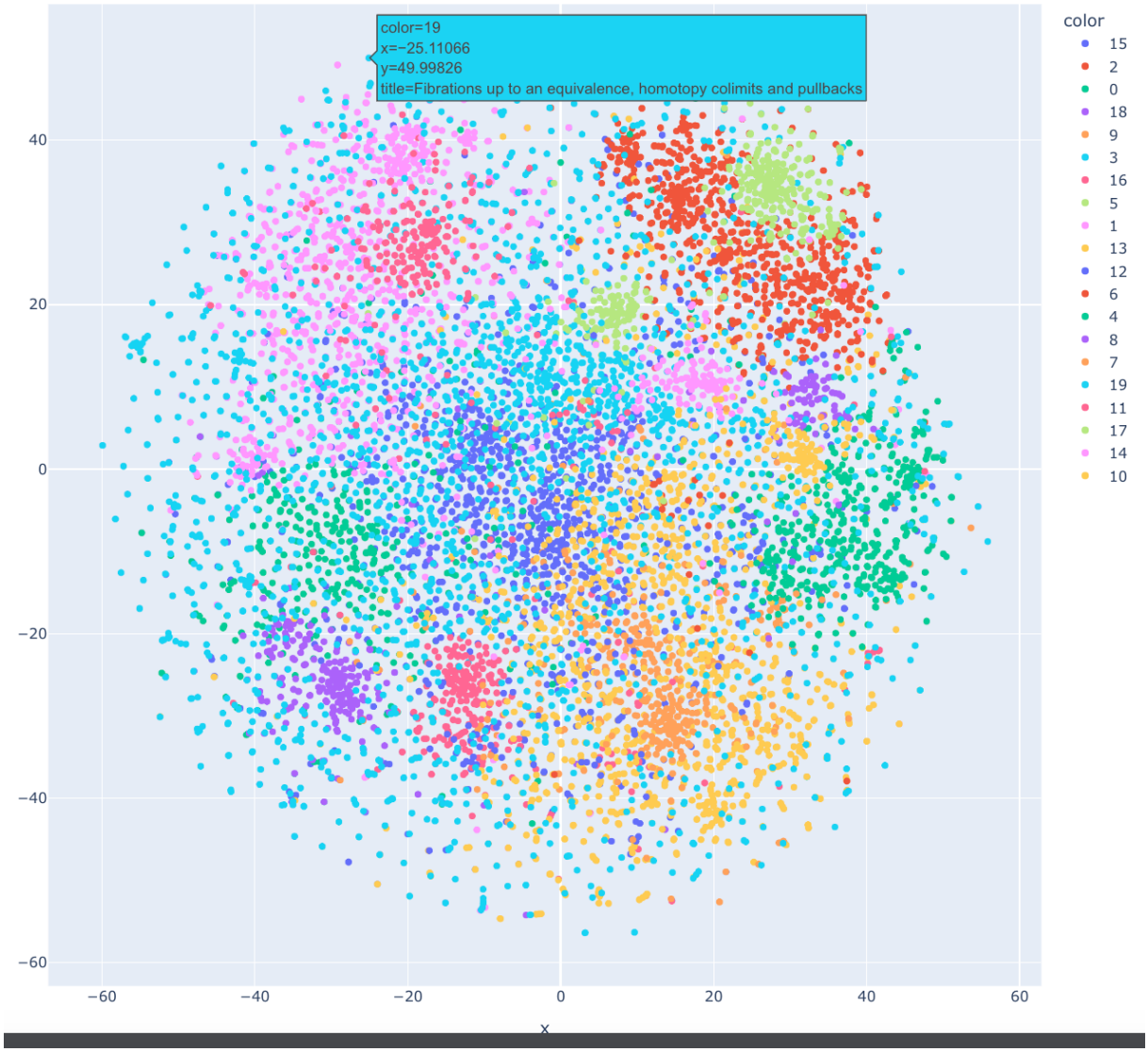

Рисунок 3.11 – Визуалізація кластерів

У оригінальному датасеті були наявні категорії. Перевіримо, які категорії були визначені належними до одного кластеру.

Зробимо це в 2 етапи. Спочатку подивимось, до яких кластерів належать різні категорії (рисунок 3.12).

	categories	cluster
0	astro-ph	[5, 15, 6, 17, 19, 11, 14, 2, 13, 3, 4, 0, 12,...
1	astro-ph gr-qc	[0, 5, 15, 19, 14]
2	astro-ph gr-qc hep-ph	[15, 14]
3	astro-ph gr-qc hep-ph hep-th	[14, 3, 4, 19, 15]
4	astro-ph gr-qc hep-th	[3, 14, 13, 19, 15]
...	...	...
1867	stat.ML cs.CV cs.LG	[0]
1868	stat.ML math.OC math.ST stat.TH	[0]
1869	stat.ML math.ST stat.TH	[18, 0, 19]
1870	stat.ML q-fin.TR	[0]
1871	stat.ML stat.ME	[0]

Рисунок 3.12 – Кластери у категоріях

Астрофізика належить до найбільшої кількості категорій. Можливою причиною є те, що ми використовуємо 20 кластерів для опису більше 2 тисяч категорій.

Пам’ятаємо, що контент-базова рекомендаційна модель може мати обмеження, яке полягає у рекомендації лише елементів однієї категорії. Застосовуючи рекомендацію по кластерам, ми таким чином можемо рекомендувати елементи з різних категорій. Також кластери не є нашим обмеженням – ми можемо використовувати і рекомендації з інших кластерів. Така гнучкість задається бізнес-правилами, які важливо підбирати під конкретний домен. У нашому випадку користувач цілком можливо знає більше однієї науки.

Ми також можемо застосувати правило, де стане необхідним рекомендувати не більше x статтей з одного кластера, якщо ми хочесто урізноманітнити рекомендації. Наприклад, це може бути застосовано, коли користувач не шукає статті по конкретній темі, а намагається побачити популярні статті з різних жанрів, щоб обрати для себе найцікавіший.

Тепер навпаки, подивимось, які категорії належать різним кластерам (рисунок 3.13).

	cluster	categories
0	0	[stat.CO math.ST stat.TH, cs.NE cs.NI, cs.MA, ...
1	1	[math.NT, math.AG, math.AT, math.GR, math.RA, ...
2	2	[astro-ph.EP astro-ph.SR, astro-ph.EP, astro-p...
3	3	[cond-mat.stat-mech hep-lat hep-th math-ph mat...
4	4	[hep-ph, hep-ph nucl-th, nucl-ex, hep-ex, hep-...
5	5	[astro-ph, astro-ph.CO, hep-th astro-ph gr-qc,...
6	6	[astro-ph.HE, astro-ph.GA, astro-ph, astro-ph....
7	7	[hep-ph nucl-th, cond-mat.mes-hall, hep-th gr-...
8	8	[hep-ex, hep-ph, hep-ph hep-ex, astro-ph.IM as...
9	9	[cond-mat.supr-con cond-mat.str-el, cond-mat.m...
10	10	[hep-ph, hep-ex, hep-ph hep-ex, hep-th, astro-...
11	11	[math.OA, math.RA, math.FA math.RA, math.GN ma...
12	12	[physics.bio-ph physics.comp-ph, hep-ph, math-...
13	13	[cond-mat.mtrl-sci cond-mat.other, nucl-th, qu...
14	14	[hep-ph, astro-ph, astro-ph.CO astro-ph.GA ast...
15	15	[math.AP, cs.IT math.IT, physics.optics, cond-...
16	16	[quant-ph, gr-qc, quant-ph astro-ph cond-mat.s...
17	17	[astro-ph.CO, astro-ph.GA astro-ph.CO, astro-p...
18	18	[cs.AI cs.LG, physics.soc-ph, cs.IT cs.MM math...
19	19	[math.RT, math.CO, math.CT math.QA, math.DG ma...

Рисунок 3.13 – Категорії у кластерах

Можна одразу виділити явні кластери – математика та статистика, фізика, комп’ютерні науки. Кластеризація допомогла виявити зв’язки між різними категоріями.

3.6 Аналіз результатів

Мета цього підрозділу полягає в застосуванні розробленої функції рекомендацій (див. рисунок 3.9) для ідентифікації та видачі схожих технічних статей. Це дозволить демонструвати практичну здатність системи аналізувати та порівнювати контент, враховуючи семантичні особливості тексту.

Спробуємо знайти рекомендації для першої статті з датасету (рисунок 3.14).

```
In [35]: recommend_df.head(1)
```

Out[35]:

	title	categories	abstract	update_date	abstract_len
0	Strichartz Estimates for Water Waves	math.AP	In this paper we investigate the dispersive ...	2010-02-02	78

Рисунок 3.14 – Таргет для рекомендацій

Ось так виглядають перші 5 рекомендацій (рисунок 3.15).

	abstract	title
77	Any finite set of linear operators on an algebra A yields an operator\nalgebra B and a module structure on A, whose endomorphism ring is isomorphic\nto a subring A^B of certain invariant elements of A. We show that if A is\na critically compressible left B-module, then the dimension of its\nself-injective hull A over the ring of fractions of A^B is bounded by the\nuniform dimension of A and the number of linear operators generating B.\nThis extends a known result on irreducible Hopf actions and applies in\nparticular to weak Hopf action. Furthermore we prove necessary and sufficient\nconditions for an algebra A to be critically compressible in the case of group\nactions, group gradings and Lie actions.\n	Irreducible actions and compressible modules
85	This work is an attempt towards a Morita theory for stable equivalences\nbetween self-injective algebras. More precisely, given two self-injective\nalgebras A and B and an equivalence between their stable categories, consider\nthe set S of images of simple B-modules inside the stable category of A. That\nset satisfies some obvious properties of Hom-spaces and it generates the stable\ncategory of A. Keep now only S and A. Can B be reconstructed ? We show how to\nreconstruct the graded algebra associated to the radical filtration of (an\nalgebra Morita equivalent to) B.\n We also study a similar problem in the more general setting of a triangulated\ncategory T. Given a finite set S of objects satisfying Hom-properties analogous\nto those satisfied by the set of simple modules in the derived category of a\nring and assuming that the set generates T, we construct a t-structure on T. In\nthe case T=D^b(A) and A is a symmetric algebra, the first author has shown that\nthere is a symmetric algebra B with an equivalence from D^b(B) to D^b(A)\nsending the set of simple B-modules to S. The case of a self-injective algebra\nleads to a slightly more general situation: there is a finite dimensional\ndifferential graded algebra B with H^i(B)=0 for i>0 and for i<<0 with the same\nproperty as above.\n	Stable categories and reconstruction
143	An important theorem in the theory of infinite dimensional Lie algebras\nstates that any affine Kac-Moody algebra can be realized (that is to say\nconstructed explicitly) using loop algebras. In this paper, we consider the\ncorresponding problem for a class of Lie algebras called extended affine Lie\nalgebras (EALAs) that generalize affine algebras. EALAs occur in families that\nare constructed from centreless Lie tori, so the realization problem for EALAs\nreduces to the realization problem for centreless Lie tori. We show that all\nbut one family of centreless Lie tori can be realized using multiloop algebras\n(in place of loop algebras). We also obtain necessary and sufficient for two\ncentreless Lie tori realized in this way to be isotopic, a relation that\ncorresponds to isomorphism of the corresponding families of EALAs.\n	Multiloop realization of extended affine Lie algebras and Lie tori
181	We say that an algebra A is periodic if it has a periodic projective\nresolution as an (A,A)-bimodule. We show that any self-injective algebra of\nfinite representation type is periodic. To prove this, we first apply the\ntheory of smash products to show that for a finite Galois covering B --> A, B\nis periodic if and only if A is. In addition, when A has finite representation\ntype, we build upon results of Buchweitz to show that periodicity passes\nbetween A and its stable Auslander algebra. Finally, we use Asashiba's\nclassification of the derived equivalence classes of self-injective algebras of\nfinite type to compute bounds for the periods of these algebras, and give an\napplication to stable Calabi-Yau dimensions.\n	Periodic resolutions and self-injective algebras of finite type
247	Universal enveloping algebras of braided m-Lie algebras and PBW theorem are\nobtained by means of combinatorics on words.\n	Universal Enveloping Algebras of Braided m-Lie Algebras

Рисунок 3.15 – Перші 5 рекомендацій

Завдяки використанню кластеризації ми мали можливість зменшити розмір датасету, у якому будемо застосовувати векторизацію. Локалізація пошуку дозволила зменшити час на підрахунок схожості, підвищуючи ефективність, що є особливо важливим у великому датасеті.

Результати показали, що векторні представлення, створені за допомогою Word2Vec, забезпечили значну точність у визначенні семантичної близькості слова, що виявилось особливо корисним у розпізнаванні контекстуальної схожості між статтями. Алгоритми кластеризації ефективно групували статті, виокремлюючи кластери з високою внутрішньокластерною схожістю і низькою міжкластерною схожістю, що свідчить про їх здатність до правильного розподілу документів за тематикою.

Незважаючи на загальний успіх, існують аспекти, які можуть бути оптимізовані. Зокрема, подальше налаштування параметрів моделей векторних представлень і вибір алгоритму кластеризації можуть сприяти підвищенню точності та ефективності системи. Також можна розглянути інтеграцію додаткових даних та методів обробки, наприклад, використання нейронних мереж для кращого врахування контексту і зменшення впливу шуму в даних.

Реалізований алгоритм демонструє високу ефективність у вирішенні поставленої задачі і може слугувати надійним інструментом для аналітичних цілей у технічних дослідженнях, а також стати основою для розвитку та удосконалення майбутніх систем обробки природної мови.

3.7 Висновки до розділу

Третій розділ кваліфікаційної роботи охоплює програмну реалізацію рекомендаційної системи, включаючи детальний аналіз та обробку датасету, використання алгоритмів Word2Vec для векторизації текстів і кластеризацію для групування даних, завершуючи аналізом результатів отриманої рекомендаційної моделі. Вибір та опис датасету надали необхідну інформацію для забезпечення релевантності та точності подальшого аналізу, зокрема визначення основних характеристик і структури даних, які важливі для точності NLP-методів.

ВИСНОВКИ

У результаті даної роботи був проведений глибокий аналіз предметної галузі, розглянуто сучасні дослідження у споріднений галузях та проведено оцінку сучасного стану речей. Робота особливо актуальна, оскільки проблема є ключовою для ефективного запуску нових продуктів і сервісів. Ефективне вирішення цієї проблеми не тільки покращує користувацький досвід, але й сприяє збільшенню лояльності та задоволення клієнтів.

У результаті практичних досліджень розроблено алгоритм, який допомагає вирішити проблему холодного старту. У процесі було розглянуто і вирішено декілька важливих викликів: великий розмір датасету, відсутність будь-яких даних про уподобання користувача, анонімізація даних та інші.

Під час вибору методів було розглянуто багато спорідненних, а також крос-модальне навчання, вивчені їх переваги і недоліки, і обрано які підходять саме до нашої предметної галузі та датасету.

Поєднання методів векторного представлення та кластеризації забезпечили швидкість та точність підрахунків, що є дуже важливим саме в сфері вирішення проблеми холодного старту.

Аналіз показав, що алгоритм ефективно забезпечує високу точність рекомендацій і вдається досягти значного покращення персоналізації рекомендацій у порівнянні з традиційними методами.

Особливо важливим є потенціал використання цієї системи у навчальному процесі. Завдяки можливості аналізу наукових текстів, система може бути застосована для пошуку і відбору схожих наукових робіт, що спростить процес складання бібліографічних списків та підготовки наукових публікацій. Це може значно покращити якість науково-дослідної роботи студентів і викладачів, а також сприяти глибшому залученню студентів до наукової роботи у сфері штучного інтелекту та обробки природної мови.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Грішаєва А.М. Дослідження та застосування методів NLP для вирішення проблеми холодного старту в рекомендаційних системах. 28-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті». Зб. матеріалів форуму. Т. 6., – Харків: ХНУРЕ. 2024. – С. 64–66.

2. Conceptual view approach of Machine Learning Based Recommendation System / G. NageswaraRao та ін. Journal of University of Shanghai for Science and Technology. 2021. Т. 23, № 07. С. 1165–1173.

3. Ricci F., Rokach L., Shapira B. Introduction to Recommender Systems Handbook. Recommender Systems Handbook. Boston, MA, 2010. С. 1–39.

4. Aggarwal C. C. An Introduction to Recommender Systems. Recommender Systems. Cham, 2016. P. 1–28.

5. Ortega F., González-Prieto Á. Recommender Systems and Collaborative Filtering. Applied Sciences. 2020. Т. 10, № 20. С. 7050.

6. Falk K. Practical Recommender Systems. Manning Publications, 2019. 432 p.

7. Yuan H., Hernandez A. A. User Cold Start Problem in Recommendation Systems: A Systematic Review. IEEE Access. 2023. С. 1.

8. Ensemble Based Hybrid Recommender Systems. International Journal of Innovative Technology and Exploring Engineering. 2020. Т. 9, № 3. С. 826–833..

9. A Review of Content-Based and Context-Based Recommendation Systems / U. Javed et al. International Journal of Emerging Technologies in Learning (iJET). 2021. Vol. 16, no. 03. P. 274.

10. Rodriguez Bertorello P. M. Recommendation Engine: Semantic Cold Start. SSRN Electronic Journal. 2020.

11. Selecting Appropriate Metrics for Evaluation of Recommender Systems. International Journal of Information Technology and Computer Science. 2019. Vol. 11, no. 1. P. 14–23.

12. Tamm Y.-M., Damdinov R., Vasilev A. Quality Metrics in Recommender Systems: Do We Calculate Metrics Consistently?. RecSys '21: Fifteenth ACM Conference on Recommender Systems, Amsterdam Netherlands. New York, NY, USA, 2021.

13. Milano S., Taddeo M., Floridi L. Recommender Systems and their Ethical Challenges. SSRN Electronic Journal. 2019.

14. Recommender Systems: Legal and Ethical Issues / ed. by S. Genovesi, K. Kaesling, S. Robbins. Cham : Springer International Publishing, 2023.

15. Goel, D. S. A Comparative Study of NLP Topic Modeling Methods and Tools. International Journal for Research in Applied Science and Engineering Technology. 2019. Vol. 7, no. 6. P. 1985–1992.

16. Attri A., Rai A., Malhotra Y. An NLP Technique on Sentiment Analysis. International Journal of Innovative Technology and Exploring Engineering. 2024. Vol. 13, no. 3. P. 28–31.

17. Research on Text Classification Method Based on NLP. Advances in Computer, Signals and Systems. 2023. Vol. 7, no. 2.

18. Chen L.-C. An Improved Corpus-Based NLP Method for Facilitating Keyword Extraction: An Example of the COVID-19 Vaccine Hesitancy Corpus. Sustainability. 2023. Vol. 15, no. 4. P. 3402.

19. Cross-Modal Representation Learning / Y. Yao et al. Representation Learning for Natural Language Processing. Singapore, 2023. P. 211–240.

20. Scaling Word2Vec on Big Corpus / B. Li et al. Data Science and Engineering. 2019. Vol. 4, no. 2. P. 157–175.

21. Alian M., Al-Naymat G. Questions clustering using canopy-K-means and hierarchical-K-means clustering. International Journal of Information Technology. 2022.