Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет *Комп'ютерних наук*
(повна назва)

Кафедра *Системотехніки*
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Проєктування методів збору та аналізу даних соціальних мереж в режимі реального часу
(тема)

Виконав:
студент 2 курсу, групи СПРм-20-1
Яворовенко К.В.
(прізвище, ініціали)

Спеціальність 122 -Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма системне проектування
(повна назва освітньої програми)

Керівник проф. Міщеряков Ю. В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Гребеннік І.В.
(підпис) (прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук

Кафедра системотехніки

Рівень вищої освіти другий (магістерський)

Спеціальність 122 – комп'ютерні науки
(код і повна назва)

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

Освітня програма системне проєктування
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«____»_____ 2021 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Яворовенко Костянтину Вікторовичу
(прізвище, ім'я, по батькові)

1. Тема роботи *Проєктування методів збору та аналізу даних соціальних мереж в режимі реального часу*

затверджена наказом університету від 8 листопада 2021 р. № 1664СТ

2. Термін подання студентом роботи до екзаменаційної комісії 18 грудня 2021 р.

3. Вихідні дані до роботи *Функція: Проєктування методів збору та аналізу даних соціальних мереж в режимі реального часу. Організація даних: база даних. Перелік використовуваних програмних засобів: ОС Microsoft Windows 10, інтегроване середовище програмування IntelliJ IDEA, браузер Google Chrome версії 75, постачальник хмарних технологій Google Cloud Platform. Технічне забезпечення: комп'ютер для виконання тестів з підтримкою Java та зі встановленим браузером Google Chrome версії 41 та вище й не менш, ніж 1Гб оперативної пам'яті.*

4. Перелік питань, що потрібно опрацювати в роботі ***4.1*** *Вступ.* ***4.2*** *Аналіз предметної області.* ***4.3*** *Постановка задачі* ***4.4*** *Огляд методів та технологій, які застосовуються.* ***4.5*** *Розробка програмної реалізації* ***4.6*** *Експериментальні дослідження.* ***4.7*** *Висновки.*

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) ***5.1*** *Діаграма конвеєру обробки даних.* ***5.2*** *Схема даних* ***5.3*** *Архітектура спроектованої системи*

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

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

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1.	*Отримання завдання на атестаційне проектування*	*14.09.21 – 20.09.21*	
2.	*Аналіз завдання, пошук літератури та аналогів з теми магістерської атестаційної роботи*	*20.09.21 – 30.09.21*	
3.	*Опрацювання літератури та аналіз проблеми збору та обробки даних соціальних мереж*	*01.10.21 – 13.10.21*	
4.	*Аналіз існуючих аналітичних систем*	*14.10.21 – 27.10.21*	
5.	*Розробка алгоритму по збору та обробки даних*	*28.10.21 – 06.11.21*	
6.	*Вибір програмних, мовних та технічних засобів для розробки системи*	*07.11.21 – 21.11.21*	
7.	*Розробка програмної реалізації згідно з алгоритму*	*22.11.21 – 30.11.21*	
8.	*Аналіз результатів, отриманих за допомогою програмного засобу*	*01.12.21 – 05.12.21*	
10.	*Оформлення пояснювальної записки та програмної документації*	*06.12.21 – 10.12.21*	
11.	*Оформлення графічної частини та презентаційних матеріалів комп'ютерного захисту*	*11.12.21 – 15.12.21*	
12.	*Представлення на рецензування*	*16.12.2020*	
13.	*Представлення атестаційної роботи в ДЕК*	*18.12.2020*	

Дата видачі завдання: _____ _____ 2021 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Міщеряков Ю.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Робота містить: *65* сторінок, *20* рисунків, 7 таблиць, *1* додаток, *21* джерело. Графічна частина дипломної роботи містить *5* плакатів.

Об’єктом дослідження є системи збору та перетворення великих даних.

Предметом дослідження є процес потокового збору та попередньої обробки даних з соціальних мереж для сентиментального аналізу тексту.

Мета кваліфікаційної роботи – проєктування ефективних методів збору та аналізу даних соціальних мереж в режимі реального часу.

Результатом кваліфікаційної роботи є метод збору та агрегації великих даних в реальному часі оснований на конвеєрній та паралельній обробці даних. Розроблене програмне забезпечення, що базується на хмарних технологіях та дозволяє виконувати горизонтальне та вертикальне масштабування.

Область застосування системи – користувачі соціальних мереж, бізнес-аналітики, SMM-спеціалісти, система орієнтована на в основному на комерційне використання з аналізу великих об’ємів даних в реальному часі.

СОЦІАЛЬНІ МЕРЕЖІ, СИСТЕМИ ЗБОРУ ДАНИХ В РЕАЛЬНОМУ ЧАСІ, ТРАНСФОРМАЦІЯ ДАНИХ, ХМАРНІ ТЕХНОЛОГІЇ, АНАЛІТИЧНА СИСТЕМА, TWITTER, ВІАЗУЛІЗАЦІЯ ВЕЛИКИХ ДАНИХ.

ABSTRACT

Thesis contains: *65* pages, *20* images, *7* tables, *1* application, *21* sources. Graphic part of the thesis contains *5* posters.

The object of research is systems for collecting and converting big data.

The subject of the study is the process of streaming and pre-processing of data from social networks for sentimental text analysis.

The purpose of the qualification work is to design effective methods of collecting and analyzing social network data in real time.

The result of the certification work is a method of collecting and aggregating real-time big data based on pipeline and parallel data processing. Developed software based on cloud technologies and allows you to perform horizontal and vertical scaling.

Scope of the system – social network users, business analysts, SMM specialists, the system is focused mainly on commercial use in the analysis of large amounts of data in real time.

SOCIAL NETWORKS, REAL-TIME DATA COLLECTION SYSTEMS, DATA TRANSFORMATION, CLOUD TECHNOLOGIES, ANALYTICAL SYSTEM, TWITTER, BIG DATA VISUALIZATION.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

БД – база даних
API – Application Programming Interface
GCP – Google Cloud Platform
SQL – Structured Query Language (мова структурованих запитів)
HTTP – HyperText Transfer Protocol
JVM – Java Virtual Machine
URL – Uniform Resource Locator (уніфікований покажчик інформаційного ресурсу)
JSON – JavaScript Object Notation
CRUD – Create, read, update, delete – 4 основні функції управління даними: «створення, читання, оновлення і вилучення».
DS – Data Science
ELT – Extract, Load, Transform (Вилучення, завантаження, перетворення)
ETL – Extract, Transform, Load (Вилучення, перетворення, завантаження)

ВСТУП

Аналіз даних, які є доступними в соціальних мережах – є гарячою темою, яка продовжує приваблювати сучасних науковців. Соціальні мережі активно розвиваються кожен день, експерти пояснюють подібний сплеск активності насамперед зростанням популярності сервісів для комунікації серед населення. Бізнеси різних розмірів охоче використовують соціальні мережі для популяризації, реалізації своїх товарів чи послуг. Проте, не всі компанії, які створюють подібні продукти дають користувачам інструменти для аналітики ефективності рекламних компаній, трендів, які цікавлять користувачів [1].

У цих випадках мають місце системи для аналізу соціальних мереж в реальному часі. Такі системи мають масу переваг. Одними із них є збір потокових даних, які з'являються кожної секунди у великих об'ємах, доступна вартість вилучених та відфільтрованих даних, обумовлена високою конкуренцією наявність підсистем з моделями машинного навчання, можливість візуалізації результатів для користувача, який не має технічного бекграунду у зрозумілому вигляді.

Тому аналітичні сервіси є доволі актуальними сьогодні, такі сервіси дозволяють використовувати дані для зросту продаж, проведення наукових досліджень.

Метою даної атестаційної роботи є дослідження існуючих методів ефективного вилучення, обробки, зберігання та візуалізації даних із соціальних мереж в реальному часі на базі сучасних хмарних технологій.

В результаті виконання атестаційної роботи буде досліджений та розроблений набір систем, які дозволить досліднику або науковцю проводити аналіз сучасних трендів на основі даних із соціальних мереж. Дана система в автоматичному режимі проводить збір даних, їх фільтрацію, зберігання, щоб користувач отримав найсвіжішу аналітику певної тематики, яка його цікавить.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис предметної області

Соціальні мережі визначаються як веб- та мобільні Інтернет-додатки, які дозволяють створювати, отримувати доступ та обмінюватися вмістом, створеним користувачами, який є повсюдно доступним. Окрім соціальних мереж (наприклад, Twitter і Facebook), для зручності ми також будемо використовувати термін «соціальні медіа», щоб охопити дійсно прості блоги та новини, які зазвичай містять неструктурований текст і доступні через веб. Соціальні медіа особливо важливі для досліджень у сфері обчислювальної соціальної науки, яка досліджує питання з використанням кількісних методів (наприклад, обчислювальної статистики, машинного навчання та складності) та так званих великих даних для аналізу даних та імітаційного моделювання [2].

Це призвело до появи численних сервісів даних, інструментів та аналітичних платформ. Однак ця легка доступність даних соціальних мереж для академічних досліджень може значно змінитися через комерційний тиск. Крім того інструменти, доступні дослідникам, далекі від ідеальних. Вони або надають поверхневий доступ до вихідних даних, або (для неповерхневого доступу) вимагають від дослідників програмувати аналітику на такій мові, як Java.

1.2 Опис сучасних проблем

Збір і аналітика соціальних медіа є багатим джерелом проблем академічного дослідження для суспільствознавців, фахівців в області комп’ютерних наук та фінансових організацій. Сучасні проблеми включають в собі:

Збір даних — хоча дані соціальних медіа доступні через API, через комерційну цінність даних більшість основних джерел, таких як Facebook і Google, ускладнюють науковцям все більший доступ до своїх «сирих» даних; дуже мало джерел соціальних даних надають доступні дані для наукових кіл і дослідників. Служби новин, такі як Thomson Reuters і Bloomberg, зазвичай стягують плату за доступ до своїх даних. На відміну від цього, Twitter нещодавно оголосив про програму Twitter Data Grants, до якої дослідники можуть подати

заявку, щоб отримати доступ до загальнодоступних твітів та історичних даних Twitter, щоб отримати уявлення з його величезного набору даних (Twitter має понад 500 мільйонів твітів на день) [3].

Очищення даних — очищення неструктурованих текстових даних (наприклад, нормалізації тексту), особливо високочастотних потокових даних у реальному часі, все ще створює численні проблеми дослідження.

Різнорідність джерел даних — дослідники все частіше об’єднують різні джерела даних: наприклад, дані соціальних мереж, дані про ринок і клієнтів у реальному часі та геопросторові дані для аналізу.

Захист даних — після того, як ви створили ресурс «великих даних», дані повинні бути захищені, вирішені проблеми з правом власності (тобто, зберігання публічних даних суперечить більшості умов використання видавців), а користувачам надати різні рівні доступу; інакше користувачі можуть спробувати «висмоктати» всі цінні дані з бази даних.

Проведення аналітики — складний аналіз даних соціальних мереж для вивчення думки (наприклад, аналіз настроїв) все ще викликає безліч проблем через іноземні мови, іноземні слова, сленг, орфографічні помилки та природний розвиток мови.

Побудова інформаційних панелей — багато платформ соціальних мереж вимагають від користувачів писати API для доступу до каналів або аналітичних моделей програм на мові програмування, наприклад Java. Хоча ці навички є розумними для комп’ютерних науковців, вони зазвичай виходять за межі більшості дослідників (соціальних наук). Непрограмні інтерфейси необхідні для надання того, що можна назвати «глибоким» доступом до «сирих» даних, наприклад, налаштування API, об’єднання каналів соціальних мереж, поєднання цілісних джерел та розробки аналітичних моделей.

Візуалізація даних — візуальне представлення даних, за допомогою якого інформація, яка була абстрагована у деякій схематичній формі з метою чіткого та ефективного передачі інформації за допомогою графічних засобів. Враховуючи обсяг залучених даних, візуалізація стає все більш важливою.

1.3 Аналіз існуючих методів добутку знань

Дані соціальних мереж, безсумнівно, є найбільшою, найбагатшою та найдинамічнішою базою доказів людської поведінки, що відкриває нові можливості для розуміння окремих людей, груп і суспільства. Інноваційні вчені та професіонали галузі все частіше знаходять нові способи автоматичного збору, комбінування та аналізу цієї маси даних. Природно, розглянути ці новаторські програми соціальних мереж у кількох абзацах досить складно. Три ілюстративних напрямки: бізнес, біонаука та соціальні науки [4].

Першими компаніями, які почали аналізувати соціальні медіа, були компанії з роздрібної торгівлі та фінансів. Компанії роздрібної торгівлі використовують соціальні медіа, щоб використовувати впізнаваність свого бренду, покращення продукту/послуг клієнтів, рекламні/маркетингові стратегії, аналіз структури мережі, поширення новин і навіть виявлення шахрайства. У фінансах соціальні мережі використовуються для вимірювання настроїв ринку, а дані новин використовуються для торгівлі. Як ілюстрацію, Bollen et al. (2011) виміряв настрої випадкової вибірки даних Twitter, виявивши, що ціни Dow Jones Industrial Average (DJIA) корелюють із настроями Twitter 2–3 днями раніше з точністю 87,6%. Wolfram (2010) використав дані Twitter для навчання моделі регресії опорних векторів (SVR) для прогнозування цін окремих акцій NASDAQ, знайшовши «значну перевагу» для прогнозування цін на 15 хвилин у майбутньому.

У галузі біонаук соціальні медіа використовуються для збору даних про великі когорти для ініціатив зі зміни поведінки та моніторингу впливу, таких як боротьба з курінням і ожирінням або моніторинг захворювань. Прикладом є біологи з університету штату Пенсильванія (Salathé et al. 2012), які розробили інноваційні системи та методики для відстеження поширення інфекційних захворювань за допомогою новинних веб-сайтів, блогів та соціальних мереж.

Програми обчислювальних соціальних наук включають: моніторинг реакцій громадськості на оголошення, виступи та події, особливо на політичні коментарі та ініціативи; уявлення про поведінку громади; опитування в соціальних мережах груп (важко контактувати); раннє виявлення нових подій, як у Twitter. Наприклад, науковці використовують комп’ютерну лінгвістику, щоб автоматично передбачити вплив новин на сприйняття громадськістю політичних

кандидатів. Інші дослідники використовують коментарі до огляду фільмів для вивчення впливу різних підходів до виділення текстових елементів на точність чотирьох методів машинного навчання — наївного байєса, дерева рішень, кластеризації максимальної ентропії та K-середніх. Наприклад, дослідники виявили, що валове національне щастя (GNH) Facebook демонструє піки та спади відповідно до основних публічних подій у США.

2 ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

Мета роботи – це створити систему, яка оптимізує роботу дослідника при аналізі трендів соціальних мереж.

Додаток повинен автоматично збирати, фільтрувати дані із соціальної мережі через потоковий API за вказаними вхідними потребами, зберігати оброблені в дані та створювати відображення для зручного аналізу.

У системі повинна мати наступні ролі: адміністратор системи та користувач-аналітик. Нижче розглянуті детальніше їх функції.

Бізнес-функції для адміністратора системи:

– Аналіз помилок при зчитуванні повідомлень;
– Покращення інструментів фільтрації;
– Бізнес-функції системи для користувачів-аналітиків;
– Введення вхідних даних;
– Вибір параметрів для аналізу;
– Формування звітів.

На основі предметної області та бізнес процесів було визначено тип корпоративної системи, а саме – MES (Manufacturing Execution System). Цей тип системи дуже підходить системі, адже основною задачею буде постійно збирати та аналізувати дані.

Функціональні вимоги для даної системи:

– можливість користувача вибирати джерело інформації;
– можливість додавання параметрів для фільтрування;
– обробка великих об’ємів даних повинна займати мінімальний час;
– можливість моніторингу отриманих даних в реальному часі;
– можливість створення зручних для аналітики звітів;
– підтримка потокового API соціальної мережі;
– можливість використання хмарних сервісів;
– ефективний фреймворк для потокової обробки даних;
– обмін повідомленнями між компонентами системи повинен підтримувати асинхронний режим;
– підтримка сховищ великих даних.

3 ОГЛЯД МЕТОДІВ ТА ТЕХНОЛОГІЙ, ЯКІ ЗАСТОСОВУЮТЬСЯ В ПРЕДМЕТНІЙ ОБЛАСТІ

3.1 Огляд соціальних мереж

Дані соціальних мереж — це такий тип даних, який отримується із соціальних мереж (наприклад, вікі, блоги, RSS-канали та новини, тощо) у різних форматах (наприклад, XML та JSON). Сюди входять набори даних, які можна отримати у реальному часі, такі як фінансові дані, дані про транзакції клієнтів, телекомунікаційні та просторові дані [5].

Програмний доступ до соціальних медіа — служби даних та інструменти для пошуку та вилучення (текстуальних) даних із медіа соціальних мереж, вікі, RSS-каналів, новин тощо. Їх можна корисно розділити на:

Джерела даних, послуги та інструменти — де доступ до даних здійснюється за допомогою інструментів, які захищають вихідні дані або забезпечують просту аналітику. Приклади: Google Trends, SocialMention, SocialPointer і SocialSeek, які надають потік інформації, що об’єднує різні канали соціальних мереж.

Канали даних через API — де набори даних і канали доступні через програмовані API на основі HTTP і повертають дані з тегами за допомогою XML або JSON тощо. Прикладами є Wikipedia, Twitter і Facebook.

Інструменти очищення та зберігання тексту — наприклад, Google Refine і DataWrangler є інструментами для очищення даних.

Інструменти аналізу тексту — бібліотеки інструментів для аналізу даних соціальних мереж після їх очищення. В основному це інструменти обробки, аналізу та класифікації природної мови, які пояснюються нижче [6].

Інструменти трансформації — прості інструменти, які можуть перетворювати текстові введені дані в таблиці, карти, діаграми (лінії, кругові, точкові, смуги тощо), шкалу часу або навіть рух (анімація на часовій шкалі), такі як Google Fusion Tables, Zoho Reports, Tableau Громадський або багато очей IBM.

Інструменти аналізу — інструменти аналітики на соціальних даних для визначення зв’язків і побудови мереж, наприклад Gephi (з відкритим кодом) або плагін NodeXL для Excel.

Платформи соціальних медіа — середовища, які надають вичерпні дані соціальних медіа та бібліотеки інструментів для аналітики. Приклади включають: Thomson Reuters Machine Readable News, Radian 6 і Lexalytics.

Медіа-платформи соціальних мереж — платформи, які забезпечують аналіз даних та аналітику в Twitter, Facebook та широкому колі інших джерел соціальних мереж.

Платформи новин — такі, як Thomson Reuters, що надають комерційні архіви/канали новин та пов’язану аналітику.

3.2 Дані соціальних мереж

Очевидно, що існує велика кількість (комерційних) сервісів, які надають доступ до соціальних мереж (наприклад, Twitter, Facebook та Вікіпедія) та служб новин (наприклад, Thomson Reuters Machine Readable News).

Дослідники постійно знаходять нові та інноваційні джерела даних для об’єднання та аналізу. Тому, розглядаючи аналіз текстових даних, потрібно враховувати кілька джерел (наприклад, соціальні мережі, RSS-канали, блоги та новини), доповнені числовими (фінансовими) даними, телекомунікаційними даними, геопросторовими даними і, можливо, даними мови та відео. Використання кількох джерел даних, безумовно, майбутнє аналітики.

Загалом дані поділяються на:

– Історичні набори даних — раніше накопичені та збережені соціальні/новинні, фінансові та економічні дані.

– Стрічки в режимі реального часу — потоки даних із потокових соціальних мереж, служб новин, фінансових бірж, телекомунікаційних послуг, пристроїв GPS та мовлення.

– Необроблені дані – необроблені комп’ютерні дані прямо з джерела, які можуть містити помилки або не проаналізовані.

– Очищені дані — виправлення або видалення помилкових (брудних) даних, спричинених розбіжностями, помилками введення ключів, відсутніми бітами, викидами тощо.

– Дані з доданою цінністю — дані, які були очищені, проаналізовані, позначені та доповнені знаннями.

Чотири найбільш поширені формати, які використовуються для розмітки тексту: HTML, XML, JSON і CSV:

– HTML — мова розмітки гіпертексту (HTML), як добре відома, є мовою розмітки для веб-сторінок та іншої інформації, яку можна переглянути у веб-браузері. HTML складається з елементів HTML, які містять теги, укладені в кутові дужки (наприклад,), у вмісті веб-сторінки.

– XML — Extensible Markup Language (XML) — мова розмітки для структурування текстових даних за допомогою <tag>…<\tag> для визначення елементів.

– JSON—JavaScript Object Notation (JSON) — це текстовий відкритий стандарт, розроблений для обміну даними, що читається людиною, і є похідним від JavaScript.

– CSV — файл значень, розділених комами (CSV), містить значення в таблиці як ряд рядків ASCII, організованих таким чином, що значення кожного стовпця відокремлюється комою від значення наступного стовпця, і кожен рядок починається з нового рядка.

HTML і XML є так званими мовами розмітки (розмітки та вмісту), які визначають набір простих синтаксичних правил для кодування документів у форматі, який читається людиною та машиною. Розмітка містить початкові теги (наприклад, <tag>), текст вмісту та кінцеві теги (наприклад, </tag>).

Багато джерел використовують нотацію об’єктів JavaScript (JSON), легкий формат обміну даними, заснований на підмножині мови програмування JavaScript. JSON — це незалежний від мови текстовий формат, який використовує конвенції, знайомі програмістам сімейства мов C, включаючи C, C++, C#, Java, JavaScript, Perl, Python та багато інших. Основними типами JSON є: Number, String, Boolean, Array (впорядкована послідовність значень, розділених комами та укладеними в квадратні дужки) та Object (невпорядкована колекція пар ключ:значення). Формат JSON проілюстрований на рис. 1 для запиту до API Twitter у рядку «UCL», який повертає два «текстових» результати від користувача Twitter «uclnews».

```
{
  "page":1,
  "query":"UCL",
  "results":[
    {
      "text":"UCL comes 4th in the QS World University Rankings. Good eh? http://bit.ly/PlUbsG",
      "date":"2012-09-11",
      "twitterUser":"uclnews"
    },
    {
      "text":"@uclcareers Like it!",
      "date":"2012-08-07",
      "twitterUser":"uclnews"
    }
  ],
  "results_per_page":2
}
```


Рисунок 3.1 – Приклад формата відповіді на запит до API Twitter

3.3 Головні концепції збору, обробки та аналізу даних

Поширеною проблемою, з якою стикаються організації, є збір даних із кількох джерел у кількох форматах. Зібрані дані необхідно перемістити в одне або кілька сховищ даних. Часто формат даних може відрізнятись від формату, який може зберегти БД, тому дані повинні бути трансформовані або очищені перед завантаженням в кінцеве сховище даних.

За кілька років для вирішення цих проблем було розроблено багато засобів, служб та процесів. Незалежно від використовуваного процесу, існує загальна необхідність координувати роботу та застосувати певний рівень перетворення даних у конвеєрі даних.

3.3.1 Вилучення, перетворення та завантаження (ETL)

Вилучення, перетворення та завантаження (ETL) – це конвеєр даних, який використовується для отримання даних із різних джерел. Потім дані перетворюються відповідно до бізнес-правил і завантажуються в цільове сховище даних. Перетворення, яке виконується в ETL, виконується в спеціалізованій підсистемі і часто включає використання проміжних таблиць для тимчасового зберігання даних при їх перетворенні та остаточному завантаженні в кінцеве сховище даних [7]. На рисунку 3.1 зображено схему потоку даних між різними етапами процесу.

Зазвичай у процесі перетворення даних застосовуються різні операції (наприклад, фільтрація, сортування, агрегування, об'єднання, очищення, дедуплікація та перевірка даних).

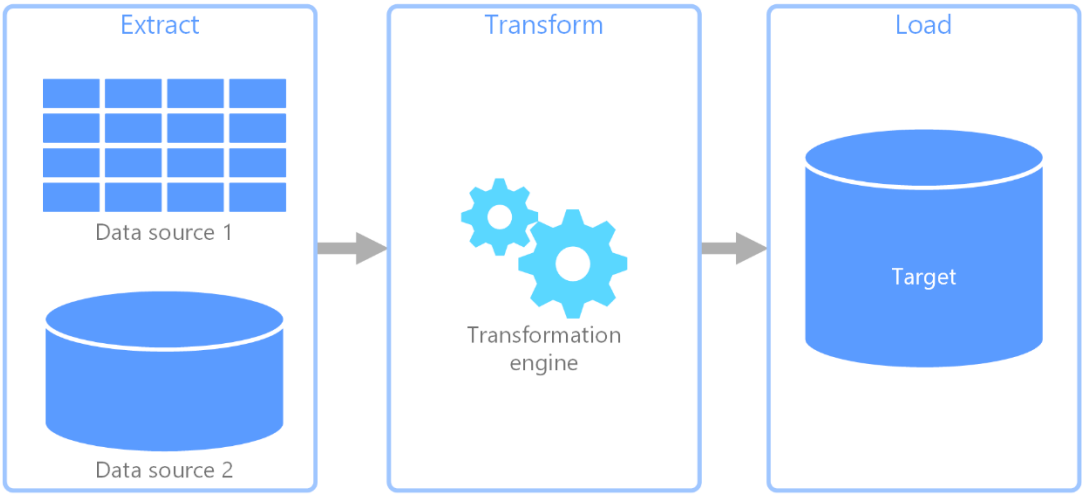

Рисунок 3.2 – Схематичне представлення ETL концепції

Часто три етапи ETL виконуються паралельно, щоб заощадити час. Наприклад, при отриманні даних процес перетворення може вже обробляти отримані дані та готувати їх для завантаження, а процес завантаження може почати обробляти підготовлені дані, не чекаючи повного завершення вилучення даних.

3.3.2 Вилучення, завантаження та перетворення (ELT)

Конвеєр вилучення даних, завантаження та перетворення (ELT) відрізняється від ETL виключно середовищем виконання перетворення. У конвеєрі ELT перетворення відбувається у цільовому сховищі даних. І тут для перетворення даних замість спеціальної підсистеми використовуються засоби обробки цільового сховища даних. Це полегшує архітектуру за рахунок видалення механізму перетворення з конвеєра. Ще однією перевагою цього підходу є те, що масштабування цільового сховища даних покращує продуктивність конвеєра ELT. Тим не менш, ELT працює належним чином, тільки якщо цільова система має достатню продуктивність для ефективного перетворення даних.

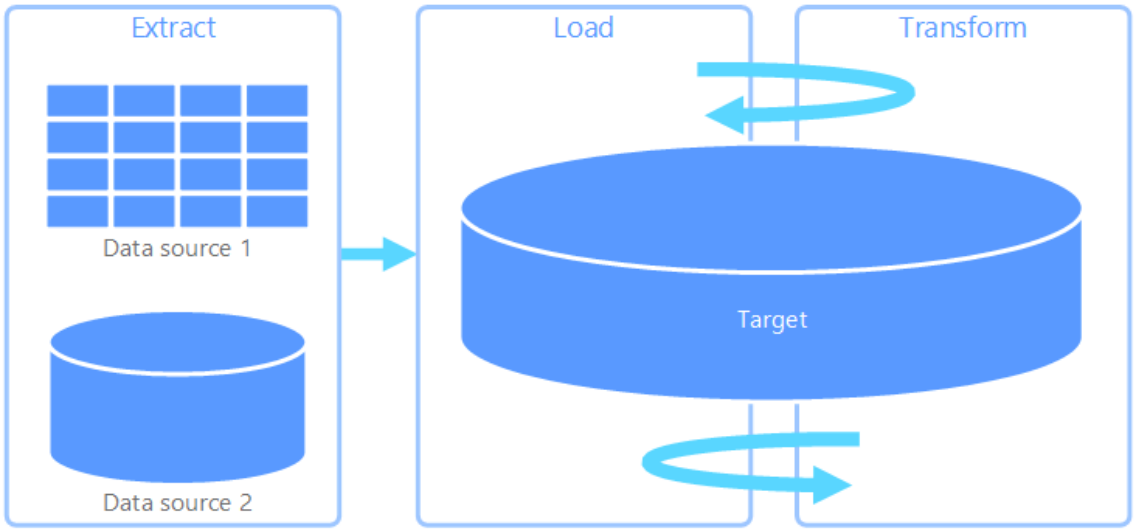

Рисунок 3.3 – Схематичне представлення ELT концепції

Зазвичай конвеєр ELT використовується для обробки великих обсягів даних. Наприклад, ви можете почати з вилучення всіх вихідних даних в неструктуровані файли в сховище, що масштабується, наприклад в розподіленій файловій системі Hadoop, сховище великих двійкових об'єктів Azure або Azure Data Lake Gen 2 (або поєднанні). Такі технології, як Spark, Hive або polybase, можна використовувати для запиту вихідних даних. Ключовою особливістю ELT є те, що сховище даних, що використовується для виконання перетворення, це те ж сховище, в якому дані в кінцевому рахунку споживаються. Це сховище даних зчитує дані безпосередньо з масштабованого сховища, замість того щоб завантажувати їх у приватне сховище. Цей підхід пропускає крок копіювання даних, представлений ETL, який часто може бути трудомісткою операцією для великих наборів даних [8].

На практиці цільове сховище даних - це сховище даних, що використовує кластер Hadoop (з використанням Hive або Spark) або виділені пули SQL в Azure синапсі Analytics. Найчастіше схема накладається на дані неструктурованих файлів під час виконання запиту і зберігається у вигляді таблиць, дозволяючи запитувати дані так само, як і будь-яку іншу таблицю в сховищі даних. Вони називаються зовнішніми таблицями, оскільки дані не знаходяться в сховищі, керованому самим сховищем даних, але в деяких зовнішніх сховищах, що масштабуються, таких як Azure Data Lake Store або сховище BLOB-об'єктів Azure.

Сховище даних керує лише схемою даних та застосовує її під час читання. Наприклад, кластер Hadoop, що використовує Hive, описує таблицю Hive, де джерелом даних є фактичний шлях до набору файлів HDFS. В Azure Polybase може досягти того ж результату - створивши таблицю для даних, що зберігаються ззовні, в самій базі даних. Коли вихідні дані завантажені, дані, наявні у зовнішніх таблицях, можна обробляти, використовуючи можливості сховища даних. У сценаріях з великими даними це означає, що сховище даних повинно мати можливість масової паралельної обробки (MPP), яка розбиває дані на невеликі фрагменти та розподіляє обробку блоків на кількох вузлах паралельно [9].

Останній етап конвеєра ELT зазвичай полягає у перетворенні вихідних даних на остаточний формат, більш ефективний для тих типів запитів, які необхідно підтримувати. Наприклад, дані можуть бути секціоновані. Крім того, ELT може використовувати оптимізовані формати зберігання, такі як Parquet, які зберігають дані, орієнтовані на рядки, в один стовпець та забезпечують оптимізоване індексування.

3.3.3 Порівняльна характеристика

В таблиці 3.1 відображено порівняльну характеристику ETL та ELT процесів.

Таблиця 3.1 – Порівняльна характеристика ETL та ELT процесів

Ознака	ETL	ELT
Концепція	Дані перетворюються на проміжному сервері, а потім зберігаються великому сховищу даних.	Дані залишаються в сховищах даних.
Доцільність використання	Інтенсивні обчислення та трансформація даних; Невелика кількість даних.	Використовується для великих обсягів даних

Продовження таблиці 3.1

Ознака	ETL	ELT
Трансформація	Трансформації виконуються в ETL-сервері/проміжній зоні.	Перетворення виконуються в цільовій системі
Час відносно завантаження	Дані спочатку завантажуються в на проміжний сервер, а потім завантажуються в цільову систему. Залежить від інтенсивності потоків даних.	Дані завантажуються в цільову систему лише один раз. Цей спосіб більш ефективніший.
Час відносно трансформація	Процес ETL повинен дочекатися завершення перетворення. Зі збільшенням розміру даних час перетворення збільшується.	У процесі ELT швидкість ніколи не залежить від розміру даних.
Час відносно управління сервером чи кластером серверів	Потребує ретельного обслуговування, оскільки вам потрібно вибрати дані для завантаження та перетворення.	Низький рівень обслуговування, оскільки дані завжди доступні.
Складність реалізації	На ранній стадії легше втілити.	Для впровадження ELT-процесу організація повинна володіти глибокими знаннями інструментів і експертними навичками.

Продовження таблиці 3.1

Ознака	ETL	ELT
Підтримка сховища даних	Модель ETL, що використовується для локальних, реляційних і структурованих даних.	Використовується в масштабованій хмарній інфраструктурі, яка підтримує структуровані, неструктуровані джерела даних.
Підтримка непомірно великих даних	Не підтримує.	Дозволяє використовувати озеро даних з неструктурованими даними.
Складність	Процес ETL завантажує лише важливі дані, визначені під час розробки.	Цей процес включає розробку від вихідних даних і завантаження лише відповідних даних.
Ціна	Високі витрати для малого та середнього бізнесу.	Низькі витрати за допомогою онлайнового програмного забезпечення як платформи обслуговування.
Агрегації	Збільшується складність із збільшенням кількості даних у наборі даних.	Потужність цільової платформи може швидко обробляти значний обсяг даних.
Обчислення	Перезаписує наявні дані.	Легко додати нове обчислення до наявної таблиці.
Апаратне забезпечення	Більшість інструментів мають унікальні вимоги до обладнання, які є дорогими.	Вартість обладнання Saas не є проблемою.

3.4 Головні компоненти архітектури для обробки великих даних

Архітектура для обробки великих даних дозволяє приймати, обробляти та аналізувати дані, які є занадто об'ємними або надто складними для традиційних систем баз даних. Час, коли організації починають використовувати великі дані, залежить від можливостей користувачів та їх коштів. Для деяких це можуть бути сотні гігабайт даних, а для інших – сотні терабайт. Оптимізація коштів на роботи з великими наборами даних корелюється з самим розміром даних. Найчастіше цей термін пов'язаний із значенням, яке можна витягти з наборів даних за допомогою розширеної аналітики, а не лише з розміром даних. Хоча у випадках вони зазвичай досить великі [10].

З роками ландшафт даних змінився. Крім того, з'явилися нові можливості для роботи із даними. Вартість сховища значно знизилася, тоді як кошторис на збір даних продовжує зростати. Деякі дані надходять у прискореному темпі, їх потрібно збирати і переглядати. Інші дані надходять повільніше, але у великих блоках. Вони часто містять дані журналів за десятиліття. Дослідники можуть зустрічатись із проблемою розширеної аналітики або проблемою, для вирішення якої потрібно використовувати машинне навчання. Це завдання, які архітектура обробки великих даних призначена вирішити.

Рішення для обробки великих даних зазвичай призначені для одного або кількох наступних типів робочого навантаження:

– пакетне оброблення джерел неактивних великих даних;

– обробка великих даних у динаміці як реального часу;

– інтерактивне вивчення великих даних;

– прогнозна аналітика та машинне навчання.

Використання архітектури для обробки великих даних можливе для наступних сценаріїв:

– зберігання та обробка даних в обсягах, надто великих для традиційної бази даних.

– перетворення неструктурованих даних для аналізу та створення звітів;

– запис, обробка та аналіз неприв'язаних потоків даних у режимі реального часу або з низькою затримкою;

На рисунку 3.3 показані логічні компоненти, що входять до архітектури для обробки великих даних [11]. Окремі рішення можуть не містити всі компоненти цієї схеми.

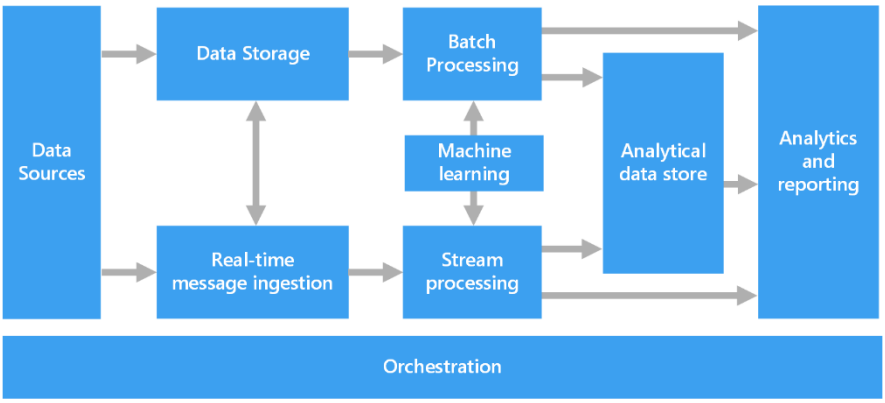

Рисунок 3.4 – Типова архітектура для обробки великих даних

3.4.1 Джерела даних

Всі рішення для обробки великих даних починаються з одного або кількох джерел даних. Приклади наведені нижче:

– Сховища даних додатків, наприклад, реляційні бази даних;

– Статичні файли, які створюються програмами, наприклад, файли журналу веб-сервера;

– Джерела даних із передачею в режимі реального часу, наприклад, пристрої Інтернету речей.

3.4.2 Сховище даних

Дані пакетної обробки зазвичай зберігаються в розподіленому сховищі файлів, де можуть міститися значні обсяги великих файлів у різних форматах. Цей тип сховищ часто називають озером даних. Таке сховище можна реалізувати за допомогою сервісу GCP BigQuery або контейнерів великих двійкових об'єктів у службі сховища Azure або GCP Cloud Storage.

3.4.3 Пакетне оброблення

Так як набори даних дуже великі, часто обробляються тривалі пакетні завдання. Ці завдання полягають у виконанні фільтрації, статистичній обробці та інших процесів підготовки даних до аналізу. Зазвичай до типу завдань входить читання вихідних файлів, їх обробка і запис вихідних даних у нові файли. Варіанти: Apache Beam або Flink, виконання завдань U-SQL в Azure Data Lake Analytics, використання власних завдань Hive, Pig або Map/Reduce у Hadoop кластері та застосування програм Java, Scala або Python у кластері Spark [12].

3.4.4 Прийом повідомлень у реальному часі

Якщо рішення містить джерела в режимі реального часу, в архітектурі має бути передбачений спосіб збирання та збереження повідомлень у режимі реального часу для потокової обробки. Це може бути просте сховище даних із папкою, в яку вхідні повідомлення розміщуються для обробки. Але для прийому повідомлень багатьом рішенням потрібне сховище, яке можна використовувати як буфер [13]. Таке сховище має підтримувати обробку з горизонтальним масштабуванням, надійну доставку та іншу семантику черги повідомлень. Ця частина архітектури потокової передачі часто називається потоковою буферизацією. Варіанти: GCP Pub/Sub, RabbitMQ, AWS SQS або SNS та Kafka.

3.4.5 Потокове оброблення

У цьому випадку система виконує фільтрацію, статистичну обробку та інші процеси підготовки даних до аналізу, збережених повідомлень у режимі реального часу. Потім оброблені потокові дані записуються у вихідний приймач. Azure Stream Analytics надає керовану службу потокової обробки на основі постійного виконання запитів SQL для неприв'язаних потоків. Крім того, для потокової передачі можна використовувати технології Apache з відкритим кодом, наприклад Storm та Spark Streaming.

3.4.6 Сховище аналітичних даних

У багатьох рішеннях обробки великих даних дані готуються до аналізу. Потім оброблені дані структуруються відповідно до формату запитів для засобів аналітики. Сховище аналітичних даних, яке використовується для обробки таких запитів, може бути реляційною базою даних типу Kimball, як можна побачити в більшості традиційних рішень бізнес-аналітики (BI). Крім того, дані можна представити за допомогою технології NoSQL з низькою затримкою, як-от HBase чи GCP BigTable або інтерактивна база даних Hive, яка надає абстракцію метаданих для файлів даних у розподіленому сховищі. GCP Bigquery– це керована служба для зберігання великих обсягів даних у хмарі. HDInsight підтримує Interactive Hive, HBase та Spark SQL, які також можна використовувати для надання даних для аналізу.

3.4.7 Аналіз та створення звітів

Більшість рішень щодо обробки великих даних призначені для аналізу та складання звітів, що дозволяє отримати важливу інформацію. Щоб розширити можливості аналізу даних, можна включити в архітектуру шар моделювання, наприклад, модель таблиці або багатовимірного куба OLAP в Azure Analysis Services або Google Data Studio. Також можна включити підтримку самостійної бізнес-аналітики з використанням технологій моделювання та візуалізації Microsoft Power BI або Microsoft Excel. Аналіз та створення звітів також може виконуватись шляхом інтерактивного вивчення даних фахівцями з їх аналізу та обробки [14]. Для таких сценаріїв багато служб Azure підтримують функції аналітичного блокнота, наприклад Jupyter, який дозволяє користувачам застосовувати свої навички роботи з Python або R. Для великомасштабного вивчення даних можна використовувати Spark.

3.4.8 Оркестрація

Більшість рішень для обробки великих даних складаються з повторюваних робочих процесів, під час яких перетворюються вихідні дані, дані переміщуються між декількома джерелами та приймачами, оброблені дані

завантажуються в сховища аналітичних даних або результати передаються безпосередньо у звіт або на панель моніторингу [15]. Щоб автоматизувати ці робочі процеси, можна використовувати технологію оркестрації, таку як фабрика даних Azure або Apache Oozie і Sqoop.

3.5 Доступ до каналу даних через API

Для дослідників, мабуть, найкориснішими джерелами даних соціальних мереж є ті, які надають програмований доступ через API, як правило, з використанням протоколів на основі HTTP. Враховуючи їх важливість для науковців, тут ми розглядаємо окремі вікі, соціальні мережі, RSS-канали, новини тощо. Як і у Вікіпедії, популярні соціальні мережі, такі як Facebook, Twitter і Foursquare, роблять частину своїх даних доступною через API.

Хоча багато сайтів соціальних мереж надають API, не всі сайти (наприклад, Bing, LinkedIn і Skype) надають доступ до API для зняття даних. У той час як все більше соціальних мереж переходять на загальнодоступний контент, багато провідних мереж обмежують вільний доступ навіть для науковців. Наприклад, у грудні 2013 року Foursquare оголосила, що більше не дозволить приватну реєстрацію на iOS 7, і тепер у партнерстві з Gnip надає безперервний потік анонімних даних реєстрації.

Якщо брати до уваги Twitter, то налаштування облікового запису за замовчуванням залишає твіти користувачів загальнодоступними, хоча користувачі можуть захистити свої твіти та зробити їх видимими лише для своїх затверджених підписників у Twitter. Однак менше 10% усіх акаунтів Twitter є приватними. Твіти з загальнодоступних облікових записів (включаючи відповіді та згадки) доступні у форматі JSON через API пошуку Twitter для пакетних запитів минулих даних і Streaming API для даних майже в реальному часі [16].

Search API — запитайте в Twitter останні твіти, що містять певні ключові слова. Він є частиною Twitter REST API версії 1.1 (він намагається відповідати принципам проектування архітектурного стилю REST, що означає передача стану репрезентації) і вимагає авторизованої програми (з використанням oAuth, відкритого стандарту авторизації) перед отриманням результатів від API.

Streaming API — потік твітів у реальному часі, відфільтрований за ідентифікатором користувача, ключовим словом, географічним розташуванням або випадковою вибіркою.

Можна отримати останні твіти, що містять певні ключові слова, через API пошуку Twitter (частина REST API версії 1.1) за допомогою такого виклику API: https://api.twitter.com/1.1/search/tweets.json?q=APPLE та в режимі реального часу за допомогою виклику потокового API: https://stream.twitter.com/1/statuses/sample.json.

API потокової передачі Twitter дозволяє отримувати доступ до даних за допомогою фільтрації (за ключовими словами, ідентифікаторами користувачів або місцеположенням) або шляхом вибірки всіх оновлень від певної кількості користувачів. Рівень доступу за замовчуванням «Spritzer» дозволяє вибірку приблизно 1 % усіх загальнодоступних статусів, з можливістю отримати 10 % усіх статусів через рівень доступу «Gardenhose» (більш підходить для інтелекту та досліджень). У соціальних мережах Streaming API часто називають Firehose — канал синдикації, який публікує всі публічні дії в одному великому потоці. Twitter нещодавно оголосив про програму Twitter Data Grants, до якої дослідники можуть подати заявку, щоб отримати доступ до загальнодоступних твітів та історичних даних Twitter, щоб отримати уявлення з його величезного набору даних (Twitter має понад 500 мільйонів твітів на день); науково-дослідні установи та науковці не отримають рівень доступу Firehose; замість цього вони отримають лише набір даних, необхідний для їхнього дослідницького проекту.

Результати Twitter зберігаються в масиві об'єктів JSON, що містить поля. Масив JSON складається зі списку об'єктів, що відповідають наданим фільтрам, і рядка пошуку, де кожен об'єкт є твітом, а його структура чітко визначена поля об'єкта, наприклад, 'created_at' і 'from_user'. Приклад складається з результату виклику API пошуку GET Twitter через http://search.twitter.com/search.json?q=financial%20times&rpp=1&include_entities=true&result_type=mixed, де параметри вказують, що пошуковий запит це «фінансові часи», один результат на сторінку, кожен твіт повинен мати вузол під назвою «об’єкти» (тобто метадані про твіт) і список «змішаних» типів результатів, тобто включати у відповідь як популярні результати, так і результати реального часу .

3.6 Методи очищення, позначення та зберігання тексту

Неможливо переоцінити важливість співвідношення «якість проти кількості» даних у аналізі та аналітиці соціальних мереж. Насправді, багато деталей аналітичних моделей визначаються типами та якістю даних. Характер даних також вплине на базу даних та обладнання, яке використовується.

Звичайно, неструктуровані текстові дані можуть бути дуже шумними (тобто брудними). Отже, процес очищення даних є важливою сферою аналітики соціальних мереж. Процес очищення даних може включати видалення друкарських помилок або перевірку та виправлення значень щодо відомого списку об’єктів. Зокрема, текст може містити неправильно написані слова, цитати, програмні коди, зайві пробіли, зайві розриви рядків, спеціальні символи, іноземні слова тощо. Тому для досягнення якісного аналізу тексту необхідно провести очищення даних на першому етапі:

– перевірка орфографії;
– видалення дублікатів;
– пошук і заміна тексту;
– зміна регістру тексту;
– видалення пробілів і недрукованих символів з тексту;
– виправлення чисел, знаків чисел і викидів;
– фіксація дат і часу, перетворення та впорядкування стовпців, рядків і даних таблиці.

Переглянувши типи та джерела необроблених даних, ми тепер переходимо до «очищення» або «очищення» даних, щоб видалити неправильну, суперечливу або відсутню інформацію. Перш ніж обговорювати стратегії очищення даних, важливо визначити можливі проблеми з даними:

Відсутність даних — коли частина інформації існувала, але з будь-якої причини не була включена в надані вихідні дані. Проблеми виникають із:

– числовими даними, коли «пусто» або відсутнє значення помилково замінено на «нуль», який потім приймається (наприклад) як поточна ціна;

– текстові дані, коли пропущене слово (наприклад, «не») може змінити весь зміст речення.

Неправильні дані — коли частина інформації вказана неправильно (наприклад, десяткові помилки в числових даних або неправильне слово в

текстових даних) або неправильно інтерпретована (наприклад, система передбачає, що вартість валюти в $, а насправді вона в £ або за умови, що текст написано американською англійською, а не австралійською англійською).

Непослідовні дані — коли частина інформації вказана невідповідно. Наприклад, із числовими даними це може бути використанням суміші форматів для дат: 2012/10/14, 14/10/2012 або 10/14/2012. Для текстових даних це може бути так само просто, як: використання одного і того ж слова в поєднанні відмінків, змішування англійської та французької в текстовому повідомленні або розміщення латинських лапок в іншому англійському тексті.

3.7 Позначення тегами неструктурованих даних

Оскільки більшість даних соціальних мереж генерується людьми, а отже, вони неструктуровані (тобто не мають попередньо визначеної структури або моделі даних), потрібен алгоритм, щоб перетворити їх у структуровані дані, щоб отримати будь-яке уявлення. Тому неструктуровані дані потрібно попередньо обробити, позначити та потім проаналізувати, щоб кількісно оцінити/аналізувати дані соціальних мереж.

Додавання додаткової інформації до даних (тобто тегування даних) можна виконувати вручну або за допомогою механізмів правил, які шукають шаблони або інтерпретують дані за допомогою таких методів, як аналіз даних і текстова аналітика. Алгоритми використовують мовну, слухову та візуальну структуру, притаманну всім формам людського спілкування. Додавання тегів до неструктурованих даних зазвичай включає тегування даних метаданими або тегами частини мови (POS). Очевидно, що неструктурована природа даних соціальних мереж призводить до неоднозначності та нерегулярності, коли вони обробляються машиною автоматично [17].

Використання одного набору даних може дати цікаву інформацію. Однак об’єднання більшої кількості наборів даних та обробка неструктурованих даних може призвести до більш цінного розуміння, що дозволить відповісти на запитання, які були неможливими заздалегідь.

3.8 Зберігання даних

Як обговорювалося, природа даних соціальних мереж дуже впливає на дизайн бази даних і, можливо, допоміжного обладнання. Також було б дуже важливо зазначити, що кожна соціальна платформа має дуже конкретні (і вузькі) правила щодо того, як можна зберігати та використовувати відповідні дані. Їх можна знайти в Умовах надання послуг для кожної платформи.

Для повноти бази даних включають:

– Реляційна база даних — база даних, організована як набір формально описаних таблиць для розпізнавання зв’язків між збереженими елементами інформації, що дозволяє встановити більш складні відносини між елементами даних. Прикладами є бази даних SQL на основі рядків і kdb + на основі стовпців, які використовуються у фінансах.

– Бази даних NoSQL — клас системи керування базами даних (СУБД), ідентифікований за її неприхильністю до широко використовуваної моделі реляційної системи керування базами даних (СУБД). Бази даних noSQL/newSQL характеризуються як: нереляційні, розподілені, з відкритим вихідним кодом і горизонтально масштабовані.

3.9 Методи аналітики публікацій в соціальних мережах

Аналіз тональності тексту (сентимент-аналіз, англ. Sentiment analysis, англ. Opinion mining) — клас методів контент-аналізу в комп'ютерній лінгвістиці, призначений для автоматизованого виявлення в текстах емоційно забарвленої лексики і емоційної оцінки авторів (думок) по відношенню до об'єктів, мова про які йде в тексті.

Тональність — емоційне ставлення автора висловлювання до деякого об'єкту (об'єкту реального світу, події, процесу або їх властивостями/атрибутам), виражене в тексті. Емоційна складова, виражена на рівні лексеми або комунікативного фрагмента, називається лексичною тональністю (або лексичним сентиментом). Тональність всього тексту в цілому можна визначити як функцію (в найпростішому випадку суму) лексичних тональностей складових його одиниць (речень) і правил їх поєднання [18].

Сентимент-аналіз стосується настроїв, емоцій, почуттів — це суб’єктивні враження, а не факти. Взагалі кажучи, аналіз настроїв має на меті визначити ставлення, висловлене автором тексту або оратором, щодо теми або загальної контекстуальної полярності документа. Науковці надають вичерпну документацію щодо основ і підходів класифікації та виділення настроїв, включаючи полярність настроїв, ступінь позитивності, виявлення суб’єктивності, ідентифікацію думки, нефактичну інформацію, наявність термінів у порівнянні з частотою, POS (частини мови), синтаксис, заперечення, особливості, орієнтовані на тему, та функції, засновані на термінах, крім термінових уніграм.

Сентимент-аналіз поділяється на конкретні підзадачі:

– Контекст настроїв — щоб отримати думку, потрібно знати «контекст» тексту, який може значно відрізнятися від спеціалізованих оглядових порталів / каналів до загальних форумів, де думки можуть охоплювати спектр тем.

– Рівень настроїв — аналіз тексту можна проводити на рівні документа, речення або атрибута.

– Суб’єктивність сентименту – визначення того, чи висловлює даний текст думку чи є фактичним (тобто без висловлення позитивної/негативної думки).

– Спрямованість/полярність настроїв — визначення того, чи є думка в тексті позитивною, нейтральною чи негативною.

– Сила настроїв — визначення «сильності» думки в тексті: слабка, помірна чи сильна.

– Мабуть, найскладнішим аналізом є визначення орієнтації/полярності та сили настроїв — позитивних (чудово, елегантно, дивовижно, круто), нейтральних (гарно, добре) і негативних (жахливо, огидно, погано, кричуще, відстой) через сленг.

– Популярний підхід — призначати оцінки орієнтації/полярності (+1, 0, −1) всім словам: позитивна думка (+1), нейтральна думка (0) і негативна думка (−1). Загальна оцінка орієнтації/полярності тексту — це сума балів орієнтації всіх знайдених слів «думка». Однак у цьому спрощеному підході є різні потенційні проблеми, такі як заперечення (наприклад, у цьому продукті немає нічого, що я ненавиджу). Одним із методів оцінки орієнтації/полярності настроїв тексту є точкова взаємна інформація (PMI) – міра асоціації, що використовується в теорії інформації та статистиці.

4 РОЗРОБКА ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

4.1 Архітектурна модель

Дослідивши сучасні методи та технології, які допомагають вирішувати задачі по збору, обробки та аналізу даних, було обрано модель Каппа архітектури.

Каппа-архітектура, схема якої зображена на рисунку 4.1, була запропонована Джеєм Крепсом як альтернатива лямбда-архітектурі. Вона має такі ж основні цілі, як і лямбда-архітектура, але при цьому є важлива відмінність: всі дані проходять через один шлях з використанням системи обробки потокових даних. Варто зазначити, що недоліком лямбда-архітектури є її складність. Логіка обробки відображається у двох різних місцях – холодних та критичних шляхах, що використовують різні платформи. Це призводить до дублювання логіки обчислень та ускладнює управління архітектурою для обох шляхів.

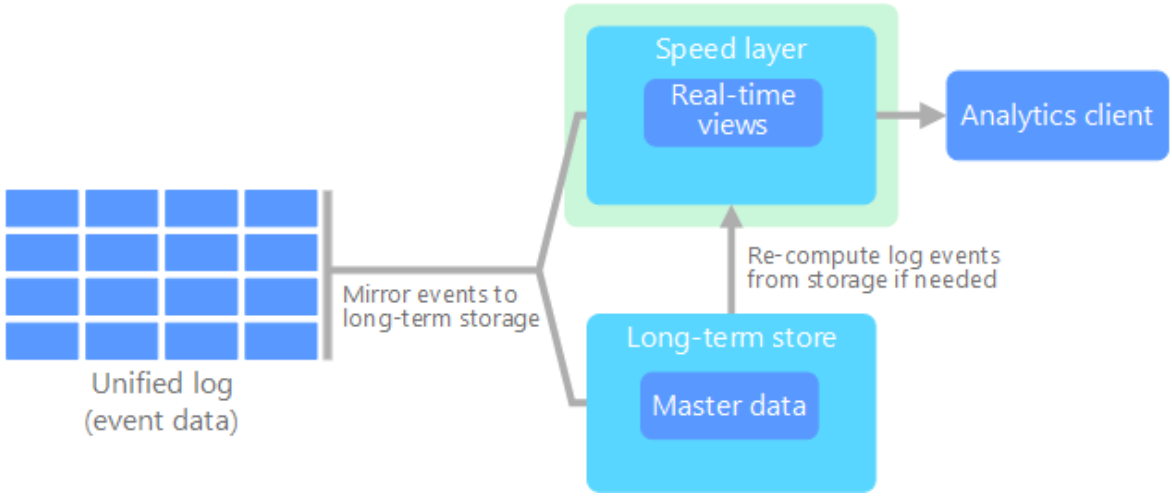

Рисунок 4.1 – Схематичне зображення Каппа-архітектури

4.2 Аналіз та обґрунтування вибору постачальника хмарних технологій

4.2.1 Сучасні постачальники хмарних технологій

На світовому ринку хмар домінують чотири постачальники хмарних послуг: Alibaba в Китаї та Азіатсько-Тихоокеанського регіону та AWS, Microsoft і Google в інших країнах.

Менші постачальники хмарних послуг в основному використовуються в сценаріях, які вимагають спеціальної підтримки застарілих служб або мають особливі вимоги до розташування, які хмарні гіганти не можуть виконати. Загалом, лідери хмарного ринку пропонують широкий спектр рішень для різних випадків використання, задовольняючи потреби високої продуктивності, доступності, безпеки та підтримки [19].

Багато великих компаній вже впровадили хмарні рішення та мають широкий спектр робочих навантажень IaaS та PaaS, включаючи розгортання додатків у хмарі. Проте деякі менші компанії перебувають у процесі міграції в хмару і шукають найкращого постачальника хмарних послуг для міграції. У цій статті будуть розглянуті сильні та слабкі сторони провідної хмарної інфраструктури та лідерів платформ, визначених компанією Gartner [20].

Згідно з порівняльною характеристикою (Рис. 4.2) головними лідерами в дані області хмарних послуг є [21]:

– Amazon Web Services;
– Azure Cloud;
– Google Cloud.

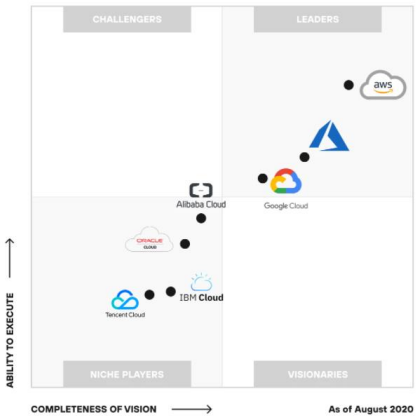

Рисунок 4.2 – Магічний квадрант постачальників хмарних послуг від Gartner

4.2.2 AWS

Дочірня компанія Amazon і заснована в 2006 році, AWS є лідером серед постачальників хмарних послуг. Це одна з перших платформ хмарних обчислень, яка стала широко доступною.

За даними Canalys, Amazon є лідером ринку IaaS, володіючи 31% частки ринку хмарних технологій. Фінансові результати за весь 2020 рік перевершили очікування компанії. AWS повідомила про дохід у 45 мільярдів доларів у 2020 році, що на 30% більше, ніж було оголошено в 2019 році.

AWS має понад 175 хмарних сервісів для широкого кола випадків використання та галузей. Найпопулярніші послуги Amazon: Amazon EC2 (обчислювальна потужність), Amazon RDS (реляційна база даних), Amazon S3 (хмарне сховище), Amazon CloudFront (сервіс доставки вмісту) і Amazon Glacier (сервіс веб-сховища). EC2 дозволяє клієнтам Amazon використовувати кластери віртуальних комп’ютерів, які доступні весь час. Більшість сервісів не доступні безпосередньо для кінцевих користувачів, однак вони надають функціональні можливості через API, які розробники можуть використовувати в додатках.

Комісія AWS залежить від вибраного обладнання та програмного забезпечення або мережевих параметрів. Абоненти можуть платити за один віртуальний комп’ютер AWS або комп’ютерний кластер. На умовах підписки Amazon забезпечує безпеку замовлених систем.

Зараз AWS обслуговує 245 країн і охоплює 25 географічних регіонів: 7 в Північній Америці, 9 в Азіатсько-Тихоокеанському регіоні, 6 в Європі, 1 в Південній Америці, 1 на Близькому Сході та 1 в Африці. Кожен регіон ізольований і складається з кількох зон доступності (загалом AWS охоплює 80 зон доступності). AWS CloudFront має 215 Edge локацій і 225 точок присутності в 90 містах у 47 країнах. Довжина хвилі AWS і локальні зони AWS забезпечують продуктивність додатків, яким для мобільних пристроїв потрібні однозначні мілісекундні затримки.

AWS має найбільшу частку ринку в IaaS і PaaS і є лідером у більшості хмарних продуктів, що пропонуються в усьому світі. Великі підприємства розгортають важливі для бізнесу робочі навантаження на Amazon частіше, ніж у будь-якого іншого постачальника хмарних послуг. Компанія має потужну мережу керованих постачальників послуг із 67 провідними консультаційними

партнерами по всьому світу. Підприємства сприймають AWS як стратегічного постачальника хмарної інфраструктури. AWS надає комплексні рішення, починаючи від серверів і закінчуючи вбудованими операційними системами в пристроях Edge, а між ними — вичерпну технологічну статистику.

Незважаючи на проголошене зниження цін, ціна на деякі послуги, як-от комп'ютерний сервіс AWS, не стала дешевшою з 2014 року. AWS оптимізує свої найкращі хмарні послуги, і якщо клієнт прив'язаний до продуктів Amazon, може бути нелегко перейти на іншого постачальника послуг. Іншою проблемою є використання менше 20% придбаних послуг, що може бути трудомістким і тривалим для реорганізації та оптимізації. Незважаючи на відповідну ціну, витрати на хмарні послуги AWS можуть бути значно високими, якщо про них не подбати.

AWS — це найзріліший і готовий до підприємств постачальник із впливовим послужним списком успіхів клієнтів, починаючи від малого та середнього бізнесу до великих підприємств. Підприємства, які використовують Amazon, постійно отримують вигоду, будучи ранніми користувачами нових послуг. Amazon Web Services пропонує понад 100 продуктів, які користувачі можуть протестувати безкоштовно. Компанія надає набір баз даних, інструментів і сервісів для розробників і мобільних пристроїв, які завжди безкоштовні. Крім того, AWS пропонує 1-річну пробну версію для певних продуктів і короткі безкоштовні пробні послуги для машинного навчання, аналітики, обчислень, безпеки та відповідності.

4.2.3 Google Cloud

Google Cloud займає третє місце в магічному квадранті постачальників хмарних послуг Gartner після AWS і Microsoft Azure. За останній рік Google Cloud значно збільшив гібридне та багатохмарне робоче навантаження за допомогою Antos, що дозволяє користувачам керувати робочими навантаженнями в Google, AWS та Azure. Крім того, Firebase, придбаний Google хмарний мобільний Backend-as-a-Service (BaaS), досить швидко зріс і став широко прийнятим розробниками. Firebase залишається дуже затребуваною платформою BaaS, незважаючи на те, що вона працює на вершині Google Cloud.

Ринкова частка Google Cloud в інфраструктурі, як ринку послуг, становить 7% на Canalys. У 2020 році Google інвестувала значні кошти в торговий персонал, що призвело до операційних збитків у розмірі 5 мільярдів доларів і загального доходу в 13 мільярдів доларів. Однак ці кроки були необхідні для масштабування бізнесу та підвищення його прибутковості.

Платформа Google Cloud пропонує 100 продуктів, які можна згрупувати в шість категорій: сховище, бази даних, обчислення та хостинг (сервери, контейнери ВМ), мережа (VPC, балансування навантаження, хмарний DNS), великі дані (Big Query - для аналізу даних, Dataflow - для пакетної та потокової обробки даних) і машинного навчання (платформа AI).

Google Cloud пропонує модель з оплатою по мірі використання. Калькулятор цін і спеціальні пропозиції можуть допомогти зрозуміти витрати на основі робочого навантаження, розташування та інших змінних. Google Cloud надає безкоштовні кредити в розмірі 300 доларів США на 90 днів для нових клієнтів, які починають працювати разом із ними. Крім того, Google Cloud має спеціальну програму для стартапів, і вони можуть отримати щонайменше 2000 доларів США кредитів для початкового запуску.

Google Cloud доступний у 200 країнах. Він охоплює 25 регіонів (9 в Північній і Південній Америці, 9 в Азіатсько-Тихоокеанському регіоні і 7 в Європі). Google Cloud має 76 зон і 144 мережевих розташування Edge. Рекомендується розгортати програми в кількох зонах в межах регіону, щоб зробити його високодоступним і стійким до відмов.

Google Cloud виділяється своїми можливостями великих даних, машинного навчання та науки про дані завдяки таким продуктам, як TensorFlow, ML Kit і Google Datasets. Він пропонує наскрізну платформу штучного інтелекту, побудовану на новітніх технологіях і підтримується такими інструментами, як TensorFlow і TPU (Tensor Processing Unit – інтегральна схема для прискорювача штучного інтелекту).

Google має проблеми з позиціонуванням як рішення IaaS корпоративного класу. Його пропозиції ще не досягли рівня зрілості підприємства, який надають AWS та Microsoft Azure, і деякі з них ще не такі досконалі, як ті, що пропонують конкуренти Google Cloud. Крім того, Google Cloud має менший пул добре обізнаних керованих постачальників послуг, ніж інші постачальники хмарних послуг.

Клієнтам Google Cloud не потрібно боятися блокування постачальників, оскільки основні пропозиції Google є відкритим кодом (Kubernetes, TensorFlow та Istio), що з часом стало галузевим стандартом. Ці послуги значно вплинули на розгортання, масштабування та керування корпоративними IT у хмарі.

4.2.4 Порівняльна характеристика хмарних сервісів для зберігання, обчислення, аналізу та візуалізації даних

В результаті аналізу сервісів та можливосте різних постачальників хмарних технологій було створено перелік типів сервісів, які потрібні задля досягнення мети атестаційної роботи. На основі переліку вимог була створена таблиця 4.1 порівняльної характеристики за наступними підгрупами:

- Регіональна доступність;
- Збереження великих даних у різному форматі;
- Сервіси аналізу великих даних;
- Інтелектуальні сервіси;
- Сервіси моніторингу;
- Інструменти для розробника.

Таблиця 4.1 – Порівняльна характеристика сервісів хмарних постачальників

Властивості та сервіси	AWS	Google Cloud	IBM Cloud	Azure	Alibaba Cloud
1	2	3	4	5	6
Регіони					
Європа	Так	Так	Так	Так	Так
Країни					
Україна	Ні	Так	Ні	Так	Ні
Сервіси збереження даних					
Бінарні неструктуровані сховища	Simple Storage Services (S3)	Google Cloud Storage	IBM Cloud Object Storage	Azure Blob Storage	Object Storage Service

Продовження таблиці 4.1

1	2	3	4	5	6
Аналітичні сервіси для великих даних					
Великі сховища даних	Amazon Redshift	Google Cloud BigQuery	Db2 Warehouse on Cloud	SQL Data Warehouse	MaxCompute
Оркестрація даних	Data Pipeline AWS Glue	Google Cloud Dataflow	Hi	Data Factory Data Catalog	DataWorks Data Integration
Обробка великих даних	Elastic MapReduce (EMR)	Google Cloud Dataproc Dataflow	IBM Analytics Engine	HDInsightC	E-MapReduce Realtime Compute
Пошук даних	Amazon Athena	Google BigQuery	SQL Query	Data Catalog Azure Data Lake Analytics Azure Search	DataWorks
Аналітика	Kinesis Data Analytics	Google Cloud Dataflow	Streaming analytics	Stream Analytics Data Lake Analytics Data Lake Store	AnalyticDB Alibaba Cloud Elasticsearch
Візуалізація	Amazon QuickSight	Google Data Studio	Hi	PowerBI (PowerBI Embedded)	DataWorks DataV
Сервіси для ДС та МН	Machine Learning SageMaker	Google Cloud AI Google Cloud Datalab Google Cloud Machine Learning Engine	Watson Machine Learning	Azure Machine Learning Studio (Azure Machine Learning Workbench)	Machine Learning Platform for AI
Інтелектуальні сервіси					
Розпізнавання мови, переклад та аналіз	Amazon Lex Amazon Comprehend Amazon Translate	Cloud Natural Language Cloud Translation API	IBM Natural Language Classifier Alchemy API Language Translator Watson Tone Analyzer	Bing Speech API LUIS - Language Understanding Intelligent Service Speaker Recognition API CRIS - Custom Recognition Intelligent Service Translator Text API Emotions API Text Analytics API	Intelligent Speech Interaction Machine Translation Quick BI Alibaba Cloud Intelligence Brain Machine Learning Platform for AI

Продовження таблиці 4.1

1	2	3	4	5	6
Моніторинг та управління					
Моніторинг і управління (DevOps)	Amazon CloudWatch AWS CloudTrail AWS X-Ray AWS Cost and Usage Report AWS Management Console	Operations Cloud Logging and Monitoring Debugger Error Reporting Google Trace	IT Operations Management IT Operations Analytics	Azure portal Azure Monitor Azure Application Insights Azure Billing API Cloud Shell Log Analytics	CloudMonitor Log Service ActionTrail Application Real-Time Monitoring Service
Інструменти розробника					
Обмін повідомленнями	Simple Queue Service (SQS) Amazon MQ	Google Cloud Pub/Sub	Compose for RabbitMQ	Azure Queue Storage Azure Service Bus Azure Event Hubs	Message Queue

4.3 Аналіз і обґрунтування вибору мови програмування

4.3.1 Порівняння мов програмування

Java є мовою комп’ютерного програмування загального призначення, яка є синхронною, заснованою на класах, об’єктно-орієнтованою та спеціально розробленою для забезпечення якомога меншої залежності від реалізації. Вона вважається ідеальною мовою для вивчення розробниками та програмістами. Час роботи компіляції програми – 1,89 секунд, а пам’яті, що використовується в секунду, – 6,01 Мб. В даний час це найпопулярніша мова програмування, і вона знову зайняла найкращу позицію в ОС Android, хоча вона трохи знизилася. Кілька років тому, Java використовувалась для мобільних додатків корпоративного рівня, для створення настільних програм та для встановлення програм Android на планшетах та смартфонах.

Таблиця 4.2 – Недоліки та переваги Java

Переваги	Недоліки
Java була розроблена таким чином, щоб бути простою у використанні, записі, компіляції, налагодженні, ніж альтернативні мови	Значно більше споживання пам'яті, ніж C або C ++.
Створює стандартну програму та код багаторазового використання	Типовий зовнішній вигляд програм графічного інтерфейсу
Має можливість простого переходу від однієї автоматичної системи обробки даних до іншої	Мова єдиної парадигми

PHP додатково називають препроцесором гіпертексту, який може бути мовою сценаріїв на стороні служби, призначеною для розробки Інтернету, проте він також використовується як мова програмування загального призначення. Інтернет-розробники повинні вивчити Hypertext Preprocessor, широко відому мову програмування.

За допомогою PHP можна надзвичайно швидко та без зусиль збільшити веб-програму. Тривалість роботи, яку використовує компілятор для виконання логіки, становить 27,64 секунд, а використовувана пам’ять – 2,57 Мб.

PHP – це фактична основа багатьох сильних систем управління вмістом, як, наприклад, Word Press.

Таблиця 4.3 – Недоліки та переваги PHP

Переваги	Недоліки
Зручне втручання	Багато ручної роботи
Швидкий доступ до бази даних	Потрібне додаткове сховище
Надзвичайно корисні варіанти текстового процесу	Немає офіційних механізмів обробки помилок

JavaScript є зрозумілою мовою програмування на високому рівні. Це мова, яка додатково характеризується як динамічна, слабко набрана, заснована на прототипі та

багатопарадигма. JavaScript надзвичайно практичний, оскільки ця мова може надзвичайно допомогти у формуванні комунікацій для вашого веб-сайту.

Можуть використовуватися різні за стилем фреймворки в JavaScript для побудови інтерфейсу користувача. Це об’єктно-орієнтована мова сценаріїв для компіляції займає 6,52 секунд пам’ять, яку використовує компілятор, становить 4,59 мб.

Надзвичайно важливо зрозуміти, що таке JavaScript для створення інтерактивних веб-сторінок. JavaScript застосовується для включення анімації на веб-сторінках, завантаження сучасних зображень, сценаріїв або об'єктів на веб-сторінку та створення високочутливих користувальницьких інтерфейсів.

Таблиця 4.4 – Недоліки та переваги JavaScript

Переваги	Недоліки
Простота створення веб-сторінки	Щоб створити щось, що навіть нагадує веб-сторінку, потрібно багато часу
Швидка передача в результаті тексту стискається	Це не так гнучко, як альтернативні розробники веб-сторінок, такі як Dreamweaver

Обґрунтування вибору мови програмування Java:

– Java має суттєві переваги перед іншими мовами та середовищами, що робить її придатною практично для будь-якого завдання програмування.

– Java легко вивчити. Java була розроблена для простоти у використанні, а тому її легко писати, компілювати, налагоджувати та вивчати, ніж інші мови програмування.

– Java є об'єктно-орієнтованою. Це дозволяє створювати модульні програми та багаторазовий код.

– Java не залежить від платформи. Однією з найважливіших переваг Java є її здатність легко переходити від однієї комп’ютерної системи до іншої. Можливість запускати одну і ту ж програму на багатьох різних системах має вирішальне значення для програмного забезпечення World Wide Web, і Java досягає цього завдяки незалежності від платформи як на вихідному, так і на двійковому рівнях.

4.4 Twitter API та структура повідомлень

API Twitter можна використовувати для програмного отримання та аналізу даних Twitter, а також для створення взаємодії в Twitter.

З роками API Twitter розширився, додавши додаткові рівні доступу для розробників та наукових дослідників, щоб мати можливість масштабувати свій доступ для покращення та дослідження публічної розмови.

Нещодавно було випущено API Twitter v2. API Twitter v2 містить сучасну основу, нові й розширені функції та швидке підключення до звичайного доступу.

В таблиці 4.5 можна побачити різні версії та рівні доступу API Twitter, та ресурси Twitter'а, які можна отримати, створити, знищити та налаштувати за допомогою API.

Таблиця 4.5 – Рівні доступу та версії API Twitter'а

Сервіс	Essential	Elevated	Academic Research
Ціна	Безкоштовно	Безкоштовно	Безкоштовно
Доступ до Twitter API v2	Підтримується	Підтримується	Підтримується
Доступ до standard v1.1	Не підтримується	Підтримується	Підтримується
Доступ до premium v1.1	Не підтримується	Підтримується	Підтримується
Доступ до enterprise	Не підтримується	Підтримується	Підтримується
Ліміт на проєкти	1 проєкт	1 проєкт	1 проєкт
Ліміт на додатки	1 додаток на проєкт	3 додатка на проєкт	1 додаток на проєкт
Доступ вилучення твітів (кількість на місяць)	500 000	2 000 000	10 000 000
Ліміт правил для фільтрованого потоку	5 правил	25 правил	1000 правил
Ліміт додавання правил для фільтрованого потоку	25 запитів / 15 хвилин	50 запитів/ 15 хвилин	100 запитів / 15 хвилин
Доступ до «full-archive search Tweets»	Не підтримується	Не підтримується	Підтримується
Доступ до «full-archive Tweet counts»	Не підтримується	Не підтримується	Підтримується
Доступ до «advanced filter operators»	Не підтримується	Не підтримується	Підтримується
Доступ до Ads API	Не підтримується	Підтримується	Підтримується

Проаналізувавши потреби в проекті було обрано рівень доступу «Essential» так як він дозволяє читати твіти у потоковому та фільтрованому форматі за заданими правилами в необхідній кількості – до 500 000 повідомлень в місяць. Також була обрана 2 версія Twitter API, так як містить широкий набір типів запитів та покращену форму відповідей у порівнянні з версією 1.1.

В таблиці 4.6 міститься опис основних ресурсів, які доступні у вищезгаданому API.

Таблиця 4.6 – Опис ресурсів Twitter API v2

Ресурс	Опис
Повідомлення (Tweets)	Мільйони твітів, які відображають публічну розмову.
Користувачі (Users)	API дозволяє шукати користувачів Twitter, щоб аналізувати їх дані для кращого розуміння аудиторії.
Списки (Lists)	Підписки та операції над ними.

4.5 Обрані сервіси та технології

Для досягнення поставленої мети були обрані наступні інструменти та технології:

Google Cloud Platform — запропонований компанією Google набір хмарних служб, які виконуються на тій же самій інфраструктурі, яку Google використовує для своїх продуктів призначених для кінцевих споживачів, таких як Google Search та YouTube. Окрім інструментів для керування, також надається ряд модульних хмарних служб, таких як обчислення, зберігання даних, аналіз даних та машинне навчання. Для реєстрації потрібно мати банківську карту або банківський рахунок. GCP надає такі оточення як інфраструктура як послуга, платформа як послуга, та безсерверні обчислення.

Apache Beam — це уніфікована модель програмування з відкритим кодом для визначення та виконання конвеєрів обробки даних, включаючи ETL, пакетну та потокову (безперервну) обробку. Конвеєри Beam визначаються за допомогою одного з наданих пакетів SDK і виконуються в одному з підтримуваних Beam

(серединних серверів розподіленої обробки), включаючи Apache Flink, Apache Samza, Apache Spark і Google Cloud Dataflow.

Google Dataflow — це повністю керований сервіс для виконання конвеєрів Apache Beam в екосистемі Google Cloud Platform.

Google BigQuery — це повністю кероване безсерверне сховище даних, яке забезпечує масштабований аналіз петабайтів даних. Це платформа як послуга (PaaS), яка підтримує запити за допомогою ANSI SQL. Він також має вбудовані можливості машинного навчання. BigQuery було анонсовано у травні 2010 року та стало загальнодоступним у листопаді 2011 року.

Google DataStudio — онлайн-інструмент для перетворення даних у настроювані інформаційні звіти та інформаційні панелі, представлений Google 15 березня 2016 року як частина корпоративного пакету Google Analytics 360. У травні 2016 року Google анонсувала безкоштовну версію Data Studio для окремих осіб і невеликих команд.

Google Pub/Sub — Pub/Sub, що означає Publisher/Subscriber, дозволяє службам спілкуватися асинхронно, із затримкою близько 100 мілісекунд. Pub/Sub використовується для потокової аналітики та конвеєрів інтеграції даних для отримання та розподілу даних. Він однаково ефективний як проміжне програмне забезпечення, орієнтоване на обмін повідомленнями, для інтеграції служб або як черга для розпаралелювання завдань.

Базовий опис архітектури, яка проєктується (відображено на рис. 4.3):

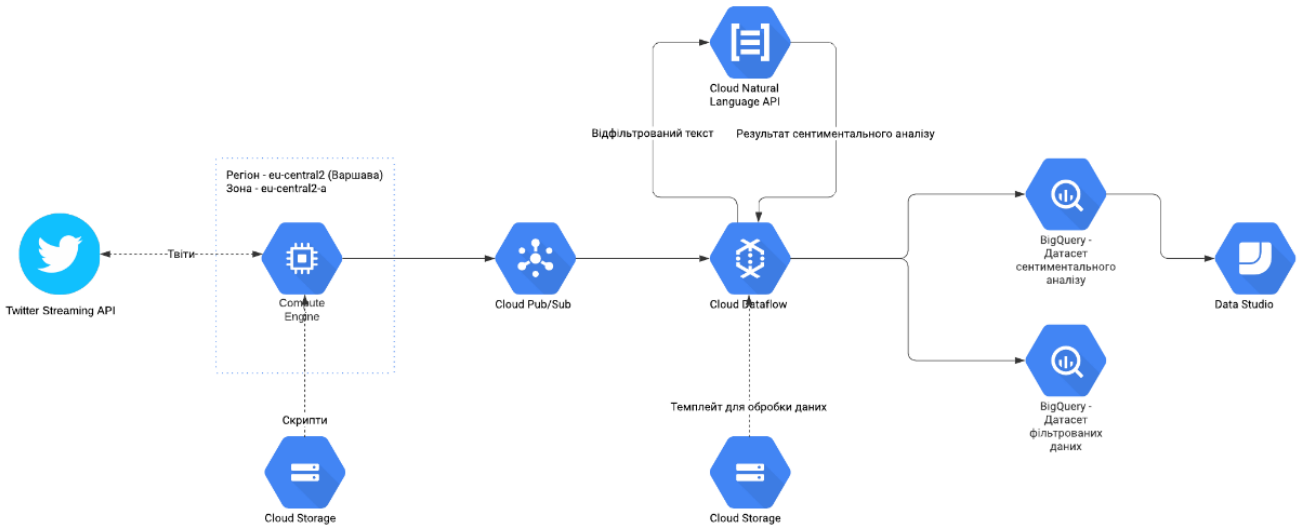

Рисунок 4.3 – Архітектура спроектованої системи

- Клієнт на віртуальній машині, який читає твіти за допомогою REST API;
- Pub/Sub для публікації твітів, потрібен для буферізації повідомлень;
- Модель ML для класифікації настроїв твітів;
- Потоковий конвеєр Apache Beam збирає твіти та класифікує їх;
- Natural Language API для сентимент-аналызу
- Зберігання результатів в BigQuery.

5 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

5.1 Створення аккаунта розробника Twitter

Під час реалізації поставленої задачі одним із перших кроків було отримання доступу до платформи розробників Twitter'у та створення додатку з назвою «diploma-twitter-streaming» (Рис. 5.1).

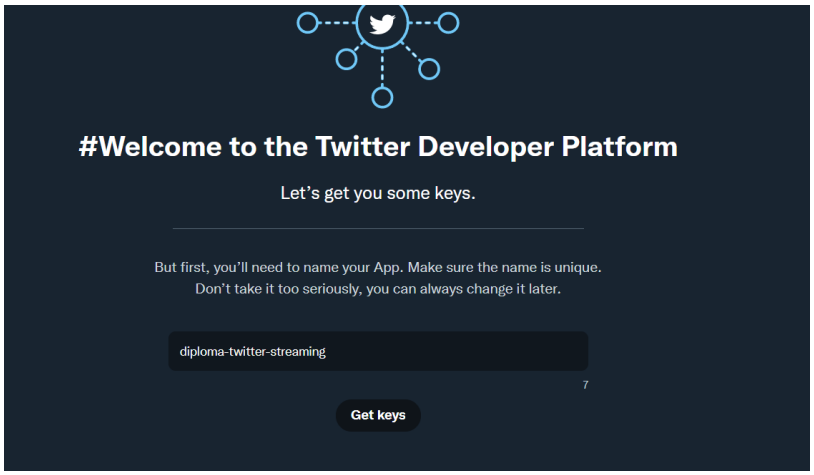

Рисунок 5.1 – Процес отримання доступу до платформи розробників

Надалі, щоб працювати з платформою були згенеровані секрети доступу до Twitter API, а саме:

– «API Key»;
– «API Key Secret»;
– «Bearer Token».

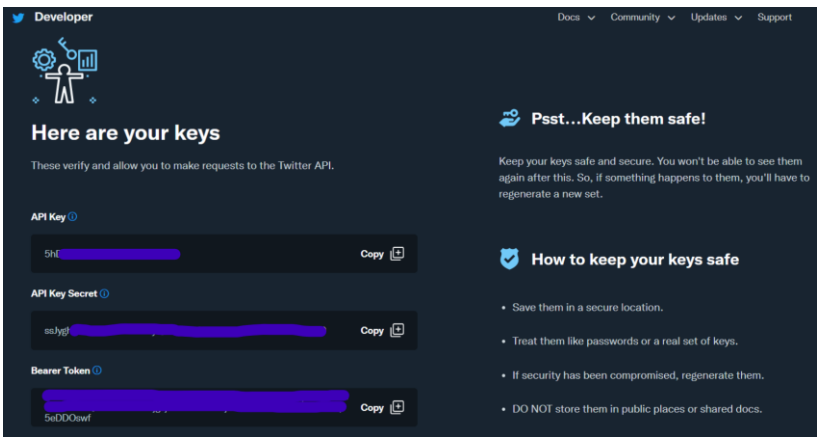

Рисунок 5.2 – Секрети доступу до Twitter API v2

Для програмного додатку, який взаємодіє з Twitter API, для виконання запитів за допомогою HTTP протоколу, використовується аутентифікація засобами API Key та API Key Secret. Для локального тестування запитів засобами curl або Postman – за допомогою Bearer Token, приклад такого запита зображено на рисунку 5.3.

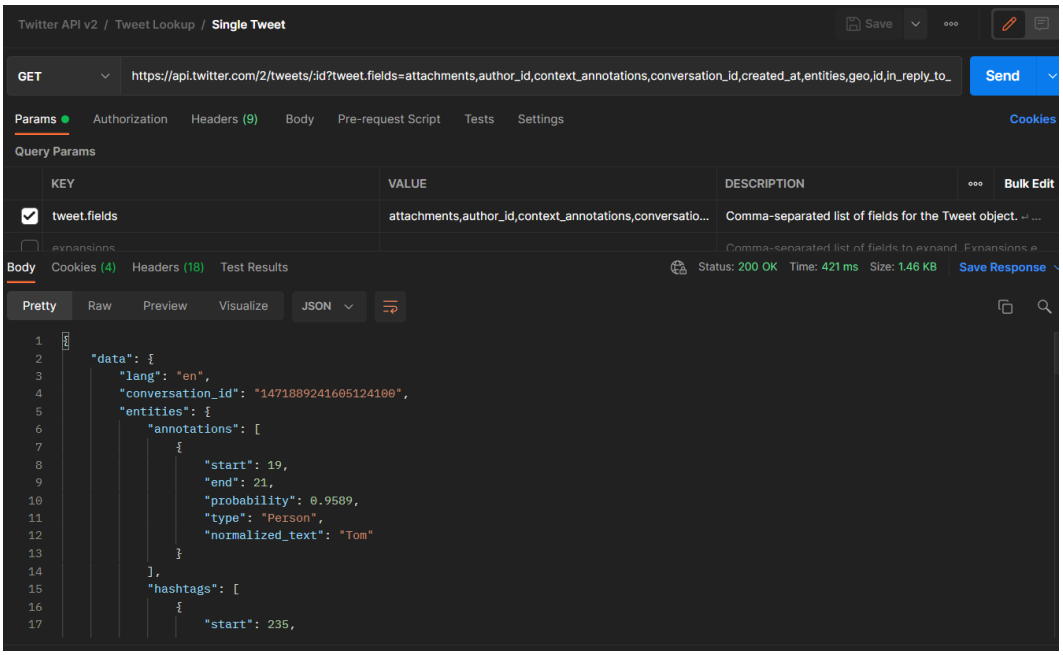

Рисунок 5.3 – Приклад запиту до api.twitter.com засобами Postman

5.1.1 API Key

API ключі можуть забезпечити:

– Авторизація проекту — щоб допомогти перевірити, чи програма, яка здійснює виклик, має доступ до її системи та якого рівня. Система також перевіряє, чи дозволене використання API у цьому проекті.

– Ідентифікація проекту — процес розпізнавання проекту або програми, яка здійснює виклик до API.

Зауважте, що ключі API не такі безпечні, як маркери, які використовуються для аутентифікації. Однак вони допомагають визначити проект або програму, яка стоїть за викликом.

Ключі генеруються в проекті, який стоїть за викликом. Це означає, що ви можете легко обмежити їх використання такими середовищами, як програми iOS

або Android. Ви також можете використовувати діапазон IP, щоб обмежити використання.

Можливість ідентифікувати проект, який здійснює виклик, означає, що ключі API можна використовувати для зв’язування інформації про використання з даним проектом.

5.1.2 Bearer Token

Bearer Token використовуються в автентифікації на основі маркерів, щоб дозволити програмі отримати доступ до API. Наприклад, програмі Calendar потрібен доступ до API Calendar у хмарі, щоб вона могла читати заплановані події користувача та створювати нові події.

Як тільки програма отримає маркер доступу, вона включатиме цей маркер як облікові дані під час виконання запитів API. Для цього він повинен передати маркер доступу до API як облікові дані Bearer у заголовку авторизації HTTP.

5.2 Конфігурація ресурсів у хмарному провайдері Google Cloud

5.2.1 Структура

Спочатку було створено проєкт з назвою «diploma-twitter-streaming» та ввімкнено необхідні API хмарного провайдера, які зображені на архітектурній діаграмі (рис. 4.3), а саме:

- Dataflow API
- Cloud Pub/Sub API
- Compute Engine API
- Cloud Monitoring API
- Cloud Logging API
- BigQuery API
- Cloud Natural Language API
- Data Studio API

5.2.2 Сервісні облікові записи

Обліковий запис для сервісу — це особливий тип облікового запису Google, призначений для представлення користувача, який не є людиною, якому потрібно пройти автентифікацію та отримати дозвіл на доступ до даних у API Google. Зазвичай облікові записи служб використовуються в таких сценаріях, як:

– Виконання робочих навантажень на віртуальних машинах (ВМ).

– Виконання робочих навантажень на локальних робочих станціях або центрах обробки даних, які викликають API Google.

– Виконання робочих навантажень, які не прив’язані до життєвого циклу людини.

– Ваша програма передбачає ідентичність облікового запису служби для виклику API Google, щоб користувачі не були безпосередньо залучені.

Для виконання практичної частини кваліфікаційної роботи було створено два сервісних акаунта (рис. 5.4), які мають можливість тільки API, до яких згідно архітектури, певні компоненти системи повинні мати доступ. Також в індивідуальному порядку можна контролювати рівень доступу, наприклад, на запис та читання, чи тільки на читання.

Рисунок 5.4 – Список сервіс акаунтів в IAM

5.2.3 Compute Engine

GCP сервіс Compute Engine може запускати загальнодоступні образи для Linux і Windows Server, які надає Google, а також приватні користувацькі снепшоти операційних систем, які моджна створювати або імпортувати із існуючих систем. Хмарний сервіс також надає змогу розгорнути контейнери Docker, які автоматично запускаються на екземплярах під керуванням загальнодоступного образу ОС, оптимізованого для контейнерів.

Варто зазначити, що користувач має право вибрати властивості машини для своїх екземплярів, наприклад кількість віртуальних ЦП і обсяг пам’яті,

використовуючи набір попередньо визначених типів машин або створюючи власні типи машин.

При імплементації практичної частина було вирішено запускати створену програму на Java, яка викликає Twitter API, трансформує отримані дані у необхідний формат та відправляє повідомлення до буферу повідомлень на віртуальній машині, яку надає Compute Engine в зоні europe-central2-a.

Конфігурація налаштована таким чином, що при створенні кожної схожої віртуальної машини будуть виконуватися Linux команди за допомогою CLI Bash, які будуть встановлювати необхідне оточення для виконання Java програми, завантажувати код з сервера контроля версії, компілювати його збирати в архів та врешті решт запускати програму. В якості типу машини було обрано f1-micro, тому що, додаток не потребує значного процесорного часу, ціна за використані ресурси буде значно нижчою.

5.2.4 Pub/Sub

Google Cloud Pub/Sub – це служба обміну повідомленнями для обміну даними про події між додатками та сервісами. Роз'єднуючи відправників і одержувачів, він забезпечує безпечний і високодоступний зв'язок між незалежно написаними програмами. Google Cloud Pub/Sub забезпечує низьку затримку та довговічний обмін повідомленнями і зазвичай використовується розробниками для впровадження асинхронних робочих процесів, розповсюдження сповіщень про події та потокової передачі даних з різних процесів або пристроїв. На рисунку 5.5 можна побачити пропускну можливість готової системи.

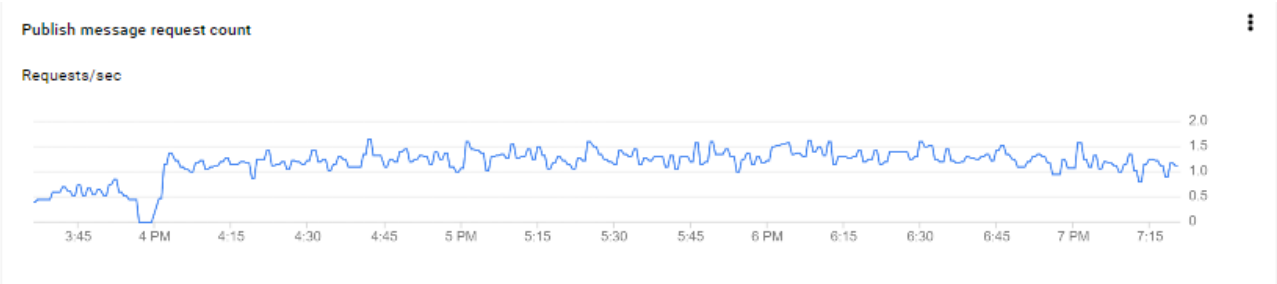

Рисунок 5.5 – Графік залежності кількості запитів за секунду

Саме для вирішення проблеми асинхронного обміну даними між Java програмою, яка зчитує повідомлення та програмою потокової обробки була використана вище згадана служба. На рисунку 5.6 зображено графік середнього об’єму повідомлення в певний момент часу. Таким чином можна відслідковувати навантаження на буфер повідомлень в режимі реального часу.

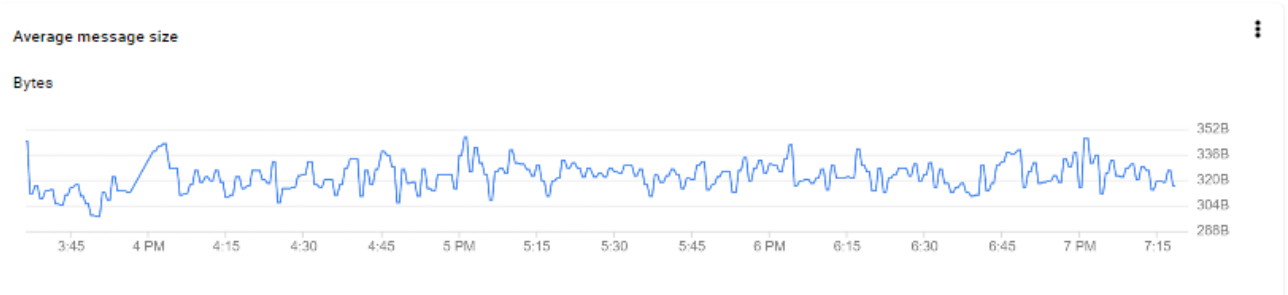

Рисунок 5.6 – Графік середнього об’єму повідомлення (Байт)

5.2.5 Dataflow

Google Cloud Dataflow — це хмарна служба обробки даних як для пакетної передачі даних, так і в режимі реального часу. Це дає змогу розробникам налаштувати конвеєри обробки для інтеграції, підготовки та аналізу великих наборів даних, таких як ті, що містяться у веб-аналітиці чи програмах для аналізу великих даних.

За допомогою цієї служби обробки даних, яка виконує завдання, що написані за допомогою бібліотек Apache Beam було автоматично розгорнуто кластер віртуальних машин (рис. 5.7).

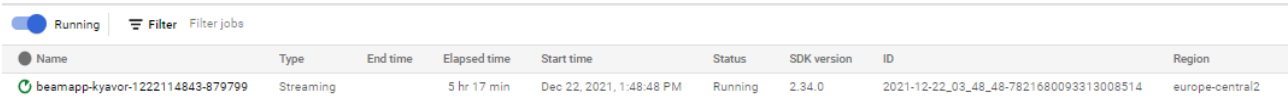

Рисунок 5.7 – Приклад запущеної служби потокової обробки

Dataflow автоматично розподіляє завдання на віртуальні машини та динамічно масштабує кластер залежно від того, як виконується завдання та яке навантаження здійснюється на систему. Це може навіть змінити порядок операцій у конвеєрі обробки, щоб оптимізувати та пришвидшити роботу.

Конфігурація конвеєра обробки потокових даних системи зображена на рисунку 5.8.

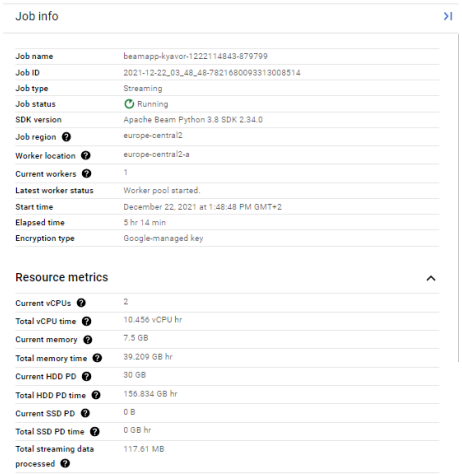

Рисунок 5.8

Повідомлення із Twitter надходять у потоковому режимі, це буквально необмежений набір даних. Тому потоковий конвеєр здається ідеальним інструментом для захоплення твітів із теми Pub/Sub та їх обробки.

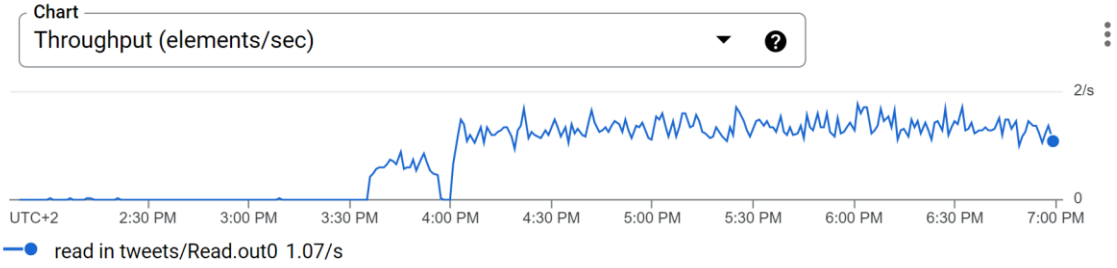

Рисунок 5.9 – Графік зчитування повідомлення

За допомогою мови програмування Python та бібліотеки Apache Beam був побудований конвеєр обробки, який складається з наступних кроків:

– Завантаження повідомлення Pub/Sub з 10-секундними інтервалами;
– Об’єднання їх у пакети по 10-20 повідомлень;
– Обробка пакету повідомлень за допомогою Cloud Natural Language API
– Агрегація результатів сентимент-аналізу тексту
– Запис результатів у відповідні таблиці даних в BigQuery

– Паралельне обрахування середнього значення сентимент-аналізу тексту в межах віконних 10-секунд і агрегація, запис даних до іншої таблиці в BigQuery

5.2.6 BigQuery

Google BigQuery — це безсерверне високомасштабване сховище даних, яке має вбудований механізм запитів. Механізм запитів здатний виконувати SQL-запити на терабайтах даних за лічені секунди і петабайти лише за хвилини. Користувачі отримують цю продуктивність без необхідності керувати будь-якою інфраструктурою та створювати або перебудовувати індекси для покращення показників швидкості та ефективності. В рамках кваліфікаційної роботи були створені необхідні об’єкти сховища даних BigQuery (рис. 5.10) з розміщенням в регіоні europe-central2.

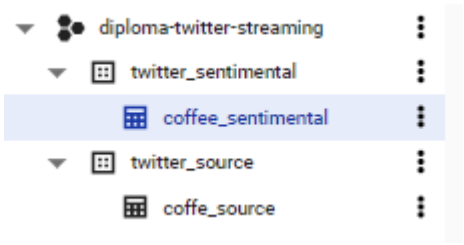

Рисунок 5.10 – Створені об’єкти сховища даних

5.2.7 Data Studio

Для візуалізації даних був обраний інструмент Data Studio, який підтримує завантаження даних із BigQuery та має великий вибір інших постачальників даних. На рисунку 5.11 зображено першу частину побудованого звіту на основі даних отриманих при обробці.

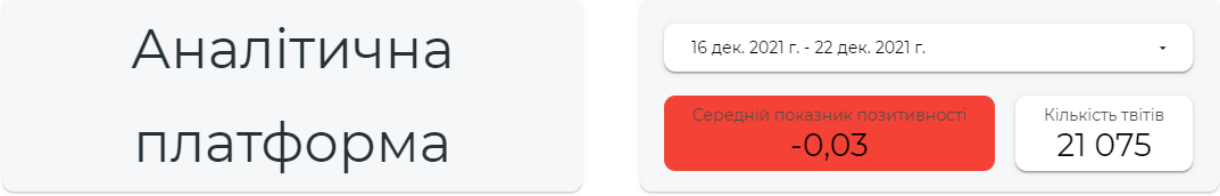

Рисунок 5.11 – Перша частина звіту в Data Studio

Data Studio — це безкоштовний інструмент, який перетворює дані користувача в інформативні, зручні для читання, доступні та гнучкі інформаційні панелі та звіти. Прикладом цього може слугувати побудована діаграма сумарного показника сентиментальності тексту за кожні 10 секунд, яка зображена на рисунку 5.12.

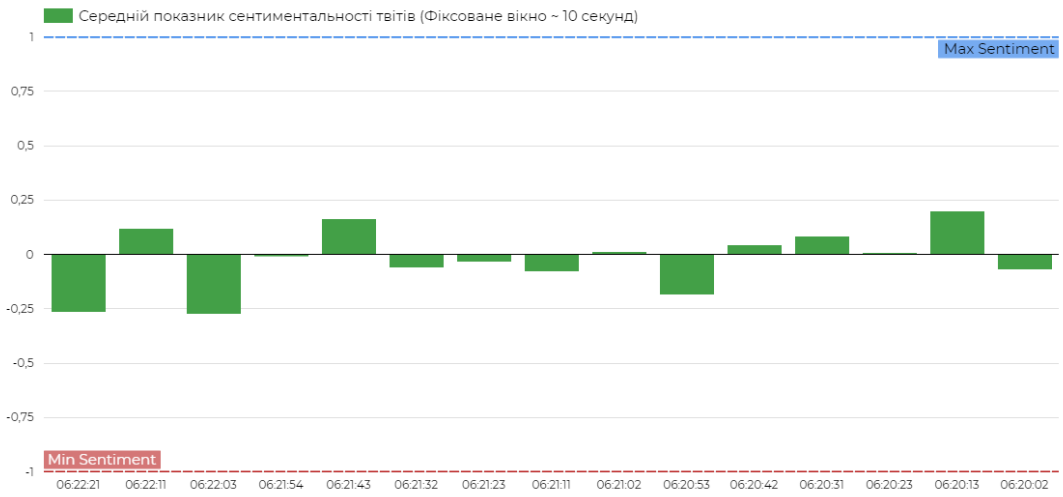

Рисунок 5.12 – Діаграма сумарного показника сентиментальності

Є різні варіанти оновлення даних для графіків та метрик: вручну або автоматично з інтервалом 15 хвилин, 1 година або 24 години.

Інструменти візуалізації даних можуть допомогти вам зрозуміти ваші дані BigQuery та проаналізувати дані в інтерактивному режимі. Ви можете використовувати інструменти візуалізації (рис. 5.13), щоб допомогти вам визначити тенденції, реагувати на них і робити прогнози на основі ваших даних.

	createdAt	sentiment ▾
1.	22 дек. 2021 г., 01:45:54	0,72
2.	22 дек. 2021 г., 01:47:22	0,7
3.	22 дек. 2021 г., 01:36:30	0,57
4.	22 дек. 2021 г., 01:37:42	0,54
5.	22 дек. 2021 г., 01:44:11	0,53
6.	22 дек. 2021 г., 01:36:05	0,5
7.	22 дек. 2021 г., 01:42:02	0,48
8.	22 дек. 2021 г., 01:52:33	0,48

	createdAt	sentiment
1.	22 дек. 2021 г., 01:53:08	-0,5
2.	22 дек. 2021 г., 05:40:33	-0,49
3.	22 дек. 2021 г., 05:01:23	-0,4
4.	22 дек. 2021 г., 02:32:13	-0,38
5.	22 дек. 2021 г., 01:48:41	-0,38
6.	22 дек. 2021 г., 05:24:04	-0,37
7.	22 дек. 2021 г., 02:35:42	-0,36
8.	22 дек. 2021 г., 06:14:53	-0,36

Рисунок 5.13 – Топ 8 найкращих та найгірших показників сентиментального аналізу опрацьованого тексту

ВИСНОВКИ

При виконанні кваліфікаційної роботи було проведено дослідження існуючих методів ефективного вилучення, обробки, зберігання та візуалізації даних із соціальних мереж в реальному часі на базі сучасних хмарних технологій. Були проаналізовані проблеми та сучасні методи здобуття даних із відкритих джерел соціальних мереж, що дало змогу обґрунтувати доцільність побудови цільової системи.

За цільову соціальну мережу було обрано Twitter, що дало змогу використовувати більш широкий набір API доступний для збору даних.

Було проаналізовано архітектури подібних систем, що дало змогу розробити більш ефективну систему, яка розширює можливості використання та представлення зібраних аналітичних даних.

В роботі запропоновано метод, що забезпечує ефективний процес збору, обробки «сирих» даних задля проведення сентимент-аналізу. Ефективність процесу забезпечується можливістю його автоматичного масштабування в хмарних сервісах.

Для користувачів запропоновано гнучкі засоби візуалізації даних, результат обробки яких, зберігається в хмарному непомірно великому сховищі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Yavorovenko K. VARIETIES OF CLOUD COMPUTING // International scientific innovations in human life. Proceedings of the 6th International scientific and practical conference. Cognum Publishing House. Manchester, United Kingdom. 2021. Pp. 228. URL:https://sci-conf.com.ua/vi-mezhdunarodnaya-nauchno-prakticheskaya-konferentsiya-international-scientific-innovations-in-human-life-15-17-dekabrya-2021-goda-manchester-velikobritaniya-arhiv/. – Дата звернення: 18.12.2021 р. – Загол. з екрану.

2. D Sitnikov, O Ryabov, I Mishcheriakov, A Kovalenko. A rough set based algebraic approach to modelling complex systems // Int. J. of Design & Nature and Ecodynamics. Vol. 13, No. 3 (2018). - 324–329 pp.

3. Google Cloud documentation [Електронний ресурс] / Google Cloud – Режим доступу: www / URL: https://cloud.google.com/docs – Дата звернення: 01.11.2021 р. – Загол. з екрану.

4. Dantas Victor. Architecting Google Cloud Solutions : Learn to design robust and future-proof solutions with google cloud technologies. [Текст] / Packt Publishing, 2021, 484 p.

5. Rehman, Mansmann, Weiler, H. Scholl. Building a data warehouse for twitter stream exploration. [Текст] / IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 2012.

6. B. Liu. Sentiment analysis and opinion mining. [Текст] / Synthesis lectures on human language technologies, 2012.

7. Akidau T., Chernyak S., Lax R. Streaming Systems. [Текст] / O'Reilly Media, Inc. 2018.

8. Han, B., Baldwin T. 2011. Lexical Normalisation of Short Text Messages. [Текст] / NAACL/HLT ’11, 368–378.

9. Bifet, A., Frank, E.: Sentiment knowledge discovery in twitter streaming data. [Текст] / Springer, Heidelberg (2010)

10. Tayal, D. K., & Yadav, S. K. (2017). Sentiment analysis on social campaign “Swachh Bharat Abhiyan” using unigram method. AI & SOCIETY, 32(4), 633-645.

11. Voorsluys W, Broberg J, Buyya R. Cloud Computing Principles and Paradigm, [Текст] / John Wiley and Sons, 2011.

12. Apache Beam Documentation [Електронний ресурс] / Apache – Режим доступу: www / URL: https://beam.apache.org/documentation/ – Дата звернення: 9.12.2021 р. – Загол. з екрану.

13. Twitter API [Електронний ресурс] / Twitter Developer Platform – Режим доступу: www / URL: https://developer.twitter.com/en/docs/twitter-api – Дата звернення: 10.12.2021 р. – Загол. з екрану.

14. Lakshmanan, V., Tigani, J. In Google BigQuery the definitive guide: Data warehousing, analytics, and machine learning at scale. [Текст] / O'Reilly Media, Inc. 2020 – 101-105 с.

15. Java web design frameworks: Review of java frameworks for web applications. [Текст] / Dr. Tejinder Singh – International Journal of Advance Research In Science And Engineering. 2015 – 592-595 с.

16. Гослінг, Джеймс. The Java Language Environment. Design Goals of the Java TM Programming Language. [Текст] / Джеймс Гослінг. – Санта-Клара: Sun Microsystems, Inc., 1996 – 548 с.

17. Tu Nguyen. Java Spring Framework in developing the Knowledge Article Management application. [Текст] / Tu Nguyen. – Helsinki: Helsinki Metropolia University of Applied Sciences, 2018 – 46 с.

18. Stefan nadschläger, Analysis of GoF Design Patterns used in Knowledge Processing Systems, Institute for Application Oriented Knowledge Processing. [Текст] / Stefan nadschläger, Josef küng. – Linz: Johannes Kepler University, 2017 – 22 с.

19. Magic Quadrant for Cloud Infrastructure and Platform Services [Електронний ресурс] / Gartner – Режим доступу: www / URL: https://www.gartner.com/doc/reprints?id=1-271OE4VR&ct=210802&st=sb – Дата звернення: 10.12.2021 р. – Загол. з екрану.

20. Top Cloud Service Providers: A Quick Comparison [Електронний ресурс] / Avenga – Режим доступу: www / URL: https://www.avenga.com/magazine/top-cloud-service-providers/ – Дата звернення: 10.12.2021 р. – Загол. з екрану.