УДК 519.713.3 DOI https://doi.org/10.32782/1995-0519.2023.5.9

ПИТАННЯ АРХІТЕКТУРИ ХМАРНОГО РІШЕННЯ ТА ІНЖЕНЕРІЇ ДАНИХ У ЗАДАЧАХ МОНІТОРИНГУ СКЛАДНОГО ТЕКСТОВОГО КОНТЕНТУ ІЗ СОЦІАЛЬНИХ МЕДІА

Георгій Чижмак
аспірант кафедри комп'ютерної інженерії та електроніки
Кременчуцький національний університет імені Михайла Остроградського, вул. Університетська, 20, Кременчук, Полтавська область, Україна, 39600, george.chizhmak@gmail.com
ORCID: 0000-0001-9284-4195

Валерій Сидоренко
кандидат технічних наук, доцент кафедри комп'ютерної інженерії та електроніки
Кременчуцький національний університет імені Михайла Остроградського, вул. Університетська, 20, Кременчук, Полтавська область, Україна, 39600, vnsidorenko@gmail.com
ORCID: 0000-0002-4449-073X

Віталій Морванюк
Senior DevOps Engineer
AirSlate Poland, вул. М. Кюрі-Склодовської, 12, Вроцлав, Республіка Польща, vitalii.morvaniuk@airslate.com
ORCID: 0009-0009-7816-5159

Тетяна Олексієнко
магістр кафедри комп'ютерної інженерії та електроніки
Кременчуцький національний університет імені Михайла Остроградського, вул. Університетська, 20, Кременчук, Полтавська область, Україна, 39600, tanuyshka2002@gmail.com
ORCID: 0009-0002-0038-350X

Сучасний етап розвитку інформаційних технологій призводить до значної трансформації в управлінні й аналізі великих обсягів текстового контенту, що походить із соціальних медіа. Спричинені великою кількістю користувачів і надмірністю інформації, яка щоденно генерується, соціальні медіа стали потужним джерелом важливих даних, які можуть бути використані для розуміння глобальних тенденцій, аналізу громадської думки та прийняття обґрунтованих рішень у різних сферах.

Ретроспективний погляд на історію Інтернету відкриває принаймні два очевидні факти: масовий перехід від повільного спілкування (електронна пошта, канали форумів) до швидшого (чати й месенджери) і значну трансформацію самої мови спілкування. Автори не мають на меті надати комплексний аналіз причин цього явища, але вважають за необхідне зупинитися на його особливостях. Специфіка спілкування в чаті призводить до того, що холдер намагається висловити свою думку коротко або дуже коротко, часто використовуючи не тільки слова, а й різноманітні символи, емодзі тощо. З іншого боку, сучасне спілкування характеризується використанням сленгу, нехтуванням орфографією, пунктуацією, а часто й моральними нормами. Ці особливості потребують створення відповідних архітектурних рішень для ефективної обробки такого роду даних.

Водночас збільшення обсягів інформації призводить до викликів у сфері моніторингу й аналізу цього текстового контенту. Використання хмарних і локальних рішень інженерії даних для моніторингу складного текстового контенту із соціальних медіа є ключовою темою дослідження.

У роботі запропоновано концепції архітектури сховищ та обробки даних, які забезпечують масштабованість, надійність і швидкодію. Водночас ці архітектурні рішення, на відміну від відомих, дають можливість створення гнучкої та ефективної системи моніторингу складного текстового контенту із соціальних медіа (з урахуванням емоджі, смайлів тощо), яка здатна пристосовуватися до змін у різних медіаплатформах і забезпечувати швидкий аналіз і відгук на актуальні події.

Ключові слова: архітектура хмарного рішення, інженерія даних, соціальні медіа, моніторинг, складний текстовий контент, data mesh, потоковий ETL, ETL, ELT, ClickHouse, Amazon Redshift, хмарні технології, сховище даних, Amazon EKS, Amazon EC2, Dask, RabbitMQ, AWS MQ, Data Pipeline.

Актуальність роботи. У сучасну епоху інформаційних конфліктів, впливу й управління інформацією набуває важливості розвиток інформаційних технологій, які дають змогу обробляти й аналізувати різноманітний контент із соціальних медіа, у тому числі й із чатів месенджерів, для розв'язання різних завдань у сферах маркетингу, реклами, соціології, державної безпеки тощо.

Поняття «різноманітний» контент, на нашу думку, потребує певного пояснення. З розвитком Інтернету і його інтерактивних можливостей суттєво змінилася як якість самого контенту (мається на увазі погіршення), так і різноманітність із семіотичного погляду: користувачі почали вільно використовувати сленг, нецензурну лексику, смайли, емодзі, аудіо, відео тощо [10, 11]. Автори в роботі [5] виділяють саме текстовий складник із цього різноманіття, ураховуючи, що більшість інформації під час спілкування передається саме нею, і вводять у цьому контексті поняття «заплутаного» текстового контенту, що певним чином звужує поняття різноманітності.

Інший бік проблеми полягає в тому, що користувачі під час спілкування в чатах, як правило, генерують короткі повідомлення, які містять певний семантичний та емоційний складники.

З урахуванням того що з погляду семантичного й сентимент-аналізу важливо дати відповіді на три основні питання: про що говорять? (семантичний складник), з якими настроями? (сентимент-складник) і хто висловив думку? (оцінювання автора думки), вищезазначені особливості тексту потребують перегляду як методів аналізу даних, так і методів їх інженерії.

Автори в роботах [5; 6; 7] розширюють і розвивають поняття безпосередньої думки [8; 9] і вводять поняття розширеної думки – структури, компонентами якої є оцінки семантичного, сентимент-складника та характеристик автора думки, отримані на основі стильометричних характеристик [6], з урахуванням того, що семантику й емоції містять ті елементи, які традиційні методи ігнорують [12; 13]: емодзі, емотікони тощо. Реалізація такого підходу на практиці потребує відповідного розвитку й питання інженерії даних.

Таким чином, у роботі робиться спроба розвити частину задачі моніторингу, пов'язану з інженерією даних, спираючись на сучасні технології: передусім із використанням доступних хмарних сервісів і мікросервісної концепції, що є одним із оптимальних рішень у частині синтезу інформаційної технології, яка має забезпечити практичну реалізацію запропонованих підходів.

Матеріал і результати досліджень. Питання архітектури та інженерії даних є важливим складником сучасних інформаційних технологій і ключовими для успіху у сфері аналізу даних, машинного навчання, штучного інтелекту та багатьох інших галузях, де дані відіграють важливу роль.

Інженерія даних оперує такими поняттями:

1. Data Pipeline (конвеєр даних) – набір елементів обробки даних, з'єднаних послідовно, де вихід одного елемента є входом наступного. Елементи конвеєра часто виконуються паралельно або з часовим інтервалом. Конвеєр може бути пакетним (історичним) або потоковим (у реальному часі). Цикл: приймання даних – обробка даних – зберігання даних – аналіз даних [2].

2. ETL (extract, transform, load) – три функції бази даних (витяг, перетворення, завантаження), які об'єднані в один інструмент для вилучення даних з однієї бази даних (БД) і розміщення їх в іншій БД [2].

3. ELT (extract, load, transform) – замість перетворення даних перед записом ELT використовує цільову систему для виконання перетворення. Однією з головних переваг ELT є скорочення часу завантаження порівняно з моделлю ETL. Використання можливостей обробки, убудованих в інфраструктуру сховищ даних (СД), скорочує час, що витрачається даними на передачу, стає більш рентабельним [3].

4. Data Warehouse (DW, сховище даних) – це база даних, оптимізована для аналізу реляційних даних, що надходять із транзакційних систем і бізнес-додатків. Структура даних і схема визначені заздалегідь для оптимізації для швидких запитів SQL, результати яких зазвичай використовуються для оперативної звітності й аналізу. Дані очищаються, збагачуються і трансформуються, щоб вони могли виступати як «єдине джерело істини», якому користувачі можуть довіряти [2].

5. Data Lake (Озеро даних) – це сховище, у якому зберігається величезна кількість необроблених даних у власному форматі, включно зі структурованими, напівструктурованими й неструктурованими даними. Структура даних і вимоги не визначені, поки дані не будуть потрібні. Озера даних забезпечують більшу гнучкість, ніж більш жорстке сховище даних [3].

Виклики, з якими стикаються застосунки для аналізу й обробки великих даних, узагальнено називають 3V, 5V, 7V і навіть більше. У цьому контексті говорять про 7 ключових викликів сучасних архітектур даних [1]:

1. Різноманітність: дані надходять у різних форматах, структурах, протоколах і розмірах із різних джерел даних. Архітектура повинна керувати диверсифікацією даних, забезпечуючи при цьому послідовний доступ до них. Вона повинна пропонувати гнучкість, а також накладати обмеження на варіації схем.

2. Швидкість: архітектура повинна керувати швидкими даними, щоб генерувати результати за коротші проміжки часу, а також повільними даними, щоб генерувати результати періодично або на вимогу. Рішення має ефективно масштабуватися по мірі зміни швидкості передачі даних.

3. Обсяг: архітектура повинна впоратися з обсягом даних, що надходять, – невеликими, великими або сплесками. Вона повинна ефективно управляти вхідними даними, а також історичними даними й пропонувати правильні варіанти для транзакційних та аналітичних варіантів використання.

4. Видимість: дані, які існують у системі, але не є видимими, настільки ж хороші (або погані), як і ті, що не існують взагалі. Архітектура повинна керувати видимістю й доступністю даних, а також їх взаємозв'язками й деталями родоводу. Вона має забезпечувати версіювання даних, щоб побачити, як набори даних змінювалися з часом, і відкотитися до певної версії або часу.

5. Достовірність – це ступінь точності, повноти й надійності даних. Дані повинні бути очищені та збагачені для забезпечення їхньої цілісності, щоб бізнес міг довіряти даним і робити впевнений аналіз.

6. Вразливість: дані повинні бути захищені від несанкціонованого доступу, де б вони не знаходилися. За даними спостерігають протягом їхнього потоку й життєвого циклу, щоб відстежити, як і ким вони використовуються.

7. Цінність: кінцевим результатом архітектури даних є уможливлення аналізу на основі даних для прийняття бізнес-рішень або створення продуктів на основі даних для покращення клієнтського досвіду.

У стандартному сценарії обробки даних програма ETL-процес вилучає певну порцію інформації з джерела, часто за заданим графіком, обробляє ці дані й потім зберігає в СД. Цей підхід можна назвати «пакетним ETL». Однак у сучасних умовах бізнесу не завжди можна чекати години або навіть дні, щоб отримати доступ до оновленої інформації. У деяких випадках необхідно отримувати дані миттєво по мірі їх створення. Підхід, який дає змогу обробляти дані в режимі реального часу, відомий як «потоковий ETL» (Streaming ETL).

У задачах моніторингу соціальних медіа потрібно швидко реагувати на зміни, тому має місце використання потокового ETL. Як показують попередні дослідження на даних мережі Livejournal, дискусії швидко згасають [5]. Тому моніторинг та обробка таких даних, наприклад, методом класифікації часових рядів сентимент оцінок [5] мають відбуватися в режимі реального часу.

У роботі запропоновано розглянути два ключові підходи до розв'язання задачі моніторингу соціальних медіа. Перший підхід базується на використанні open-source технологій і дає можливість розгортання системи як на локальних серверах, так і на організованих кластерах серверів (далі локальне рішення). Другий підхід включає використання хмарних рішень, які надаються провайдерами хмарних послуг, на прикладі AWS (далі – хмарне рішення).

Почнемо із загальної частини, яка актуальна для обох підходів, а саме з інженерії даних. Пропонується підхід data mesh [4] для створення системи зберігання й обробки даних, яка дає змогу розділити сховища та Data Pipelines для кожного соціального медіа й забезпечити однакову структуру як саме сховищ, так і Data Pipelines з можливістю адаптації до специфічних особливостей кожної медіаплатформи.

Ключовими аспектами є вибір сховища для зберігання даних та архітектури самого рішення на рівні таблиць.

Для локального рішення пропонується використовувати ClickHouse як сховища даних. ClickHouse відзначається такими ключовими особливостями, які роблять його відмінним рішенням для побудови Data Warehouse в контексті моніторингу соціальних медіа:

1. Колонкова архітектура БД: ClickHouse використовує колонкову архітектуру, що дає змогу ефективно стискати та швидко обробляти великі обсяги даних, що є надзвичайно важливим для аналізу текстових даних і численних метрик у контексті соціальних медіа.

2. Горизонтальна масштабованість: ClickHouse підтримує горизонтальне масштабування, що дає змогу додавати нові сервери для обробки даних без перерви в роботі системи, забезпечуючи високу доступність і масштабованість.

3. Підтримка SQL-запитів: використання мови SQL у ClickHouse спрощує аналітичний

процес і сприяє інтеграції з різними аналітичними інструментами.

4. Підтримка реплікації й резервування даних: ClickHouse дає можливість реплікації даних для забезпечення надійності та можливості відновлення в разі непередбачуваних ситуацій.

5. Багатий набір функцій і підтримка масштабних операцій: ClickHouse пропонує розширений набір функцій для зручної обробки даних, включаючи агрегацію, фільтрацію та обчислення.

6. Join запити: ClickHouse відзначається високою ефективністю під час обробки join запитів. Він використовує колонкову архітектуру, яка дає змогу виконувати злиття (join) даних на великій кількості стовпців швидше, ніж традиційні реляційні бази даних. Це дуже важливо для аналізу даних, де злиття декількох таблиць може бути різноманітним і складним завданням.

7. Робота з timeseries даними: ClickHouse оптимізований для роботи з часовими рядами (timeseries). Він забезпечує швидкий та ефективний доступ до даних, включаючи агрегації за часовими інтервалами, вибірки по даті й часу, а також аналітичні операції, необхідні для обробки часових даних.

8. Підтримка JSON: ClickHouse також дає можливість зберігати й обробляти дані у форматі JSON. Він уміє ефективно вибирати, оновлювати та аналізувати дані, що зберігаються в JSON-форматі. Це корисно для обробки структурованих і напівструктурованих даних, які часто зустрічаються в соціальних медіа.

Як хмарне рішення для зберігання даних обрано Amazon Redshift. Amazon Redshift є високопродуктивною колонковою базою даних, спеціально розробленою для аналітичних завдань та обробки великих обсягів даних у хмарному середовищі AWS. Ось деякі ключові переваги використання Amazon Redshift у контексті моніторингу соціальних медіа:

1. Висока продуктивність: Amazon Redshift використовує оптимізовану архітектуру для швидкого завантаження та запитів до даних. Це дає змогу виконувати складні аналітичні запити на великих обсягах даних у найкоротший термін.

2. Горизонтальне масштабування: Amazon Redshift підтримує горизонтальне масштабування, що дає можливість легко збільшувати потужність обчислень та обсяги даних, якщо це необхідно.

3. Інтеграція з іншими AWS сервісами: Amazon Redshift легко інтегрується з іншими сервісами AWS, що спрощує розгортання й управління інфраструктурою.

4. Безпека даних: Amazon Redshift надає різні засоби для забезпечення безпеки даних, включаючи можливість налаштування ролей доступу, шифрування даних і моніторингу заходів безпеки.

5. Складні аналітичні можливості: Amazon Redshift підтримує розширений набір функцій SQL, що дає змогу виконувати складні аналітичні операції та обчислення прямо в базі даних.

6. Підтримка JSON і JSONB: Amazon Redshift має вбудовану підтримку для роботи з JSON-даними, що дає можливість зберігати й обробляти дані в цьому форматі, що може бути важливо для соціальних медіа, де дані часто мають структуру JSON.

7. Споживання медіа контенту: Amazon Redshift відмінно справляється з аналізом медіа контенту, включаючи зображення та відео, завдяки можливості зберігання інформації про медіа в бінарному форматі.

8. Інструменти для адміністрування та моніторингу: AWS надає широкий спектр інструментів для адміністрування й моніторингу Amazon Redshift, що спрощує управління та підтримку бази даних.

Схема сховища для зберігання сирих даних наведена на рис. 1 і ґрунтується на архітектурі зірки (відомій також як архітектура Кімбела). Ця архітектура відзначається такими особливостями структури:

1. Таблиця фактів (Fact Table): у центрі архітектури знаходиться таблиця фактів, яка є центральною точкою для зв'язку всіх інших таблиць і зберігання агрегованих метрик.

2. Вимірювальні таблиці (Dimension Tables): поза фактичною таблицею існують вимірювальні таблиці, які містять різноманітні атрибути й додаткову інформацію щодо об'єктів, які аналізуються. Ці таблиці допомагають виконувати агрегації та фільтрації даних на основі різних характеристик.

3. Визначені ключі зв'язку: зв'язок між фактичною таблицею та вимірювальними таблицями визначається за допомогою ключів, які відображають залежність між даними. Ці ключі допомагають виконувати злиття й аналіз на основі різних атрибутів.

Така система зберігання даних використовує переваги ClickHouse й Amazon Redshift для забезпечення ефективного аналізу й обробки інформації.

Схема сховища для зберігання оброблених даних, а саме кортежів думок [7], наведено на рис. 2. Ці сховища також організовано за прин-

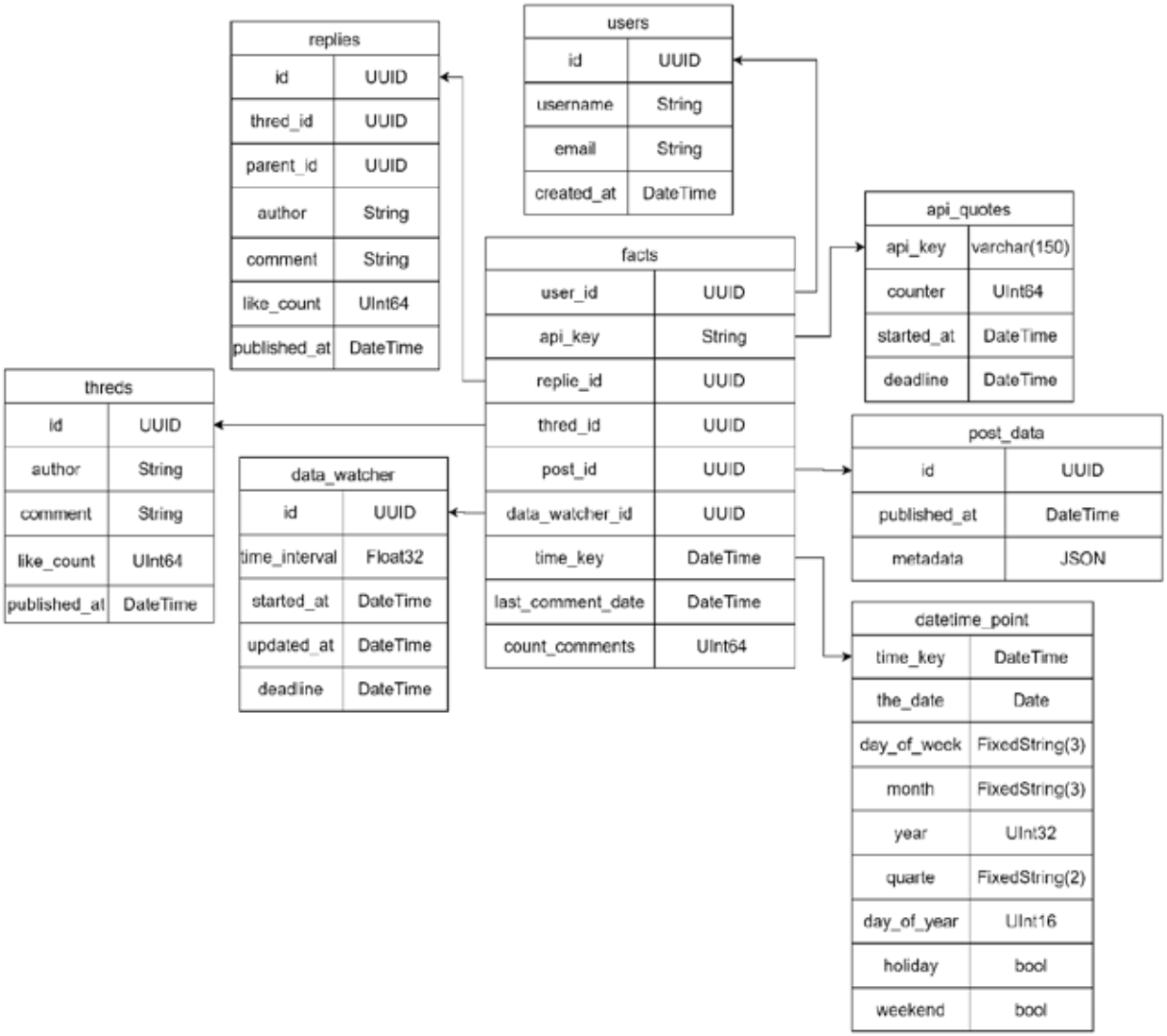

Рис. 1. Схема сховища для зберігання сирих даних

ципом data mesh і використовуються для оперативного аналізу розширених думок на основі складного текстового контенту соціальних медіа. Сховища організовані з використанням ClickHouse для локального рішення й Amazon Redshift для хмарного, складаються з однієї денормалізованої таблиці для пришвидшення обробки й аналізу даних у режимі реального часу.

Поля типу JSON дають змогу зберігати вкладені структури даних без зав'язки на жорстку структуру, що дає можливість додавати нові або оновлювати застарілі дані елементів кортежу, не змінюючи структури таблиці. Наприклад, у разі додавання нових або оновлення наявних атрибутів холдерів думок, про що свідчать роботи [6].

Спираючись на роботи [6], висновуємо, що елементи *op* кортежу думок можуть зберігатися в полі типу Array(Float32).

Поля entity та feature також мають тип JSON для зберігання вкладених ієрархічних структур.

Використання мікросервісної архітектури з Docker контейнеризацією має низку переваг у контексті побудови ELT та ETL Data Pipelines соціальних медіа, серед яких варто зазначити такі:

1. Гнучкість і модульність: мікросервіси дають змогу розділити Data Pipeline на невеликі окремі компоненти, кожен із яких відповідає за конкретну функцію. Це полегшує розвиток і модифікацію системи, оскільки дає змогу змінювати окремі компоненти без впливу на інші частини системи.

2. Швидкість розгортання та масштабування: Docker контейнери дають змогу швидко розгортати й масштабувати окремі сервіси. Це особливо важливо у випадку обробки великих обсягів

extended_opinions	
id	UUID
entity	JSON
feature	JSON
op	Array(Float32)
holder	JSON
time	DateTime
parent_id	UUID
post_id	UUID
comment_id	UUID
user_account_id	UUID

Рис. 2. Схема сховища для зберігання оброблених даних

даних, коли необхідно швидко реагувати на зміни в обсягах даних і навантаженні.

3. Ізоляція та безпека: Docker контейнери забезпечують високий рівень ізоляції між компонентами Data Pipeline, що сприяє безпеці й запобігає взаємному впливу в разі помилок або атак на систему.

4. Інтеграція з іншими інструментами: мікросервіси можуть легко інтегруватися з іншими інструментами та сервісами, що дає змогу розширити можливості Data Pipeline й підключити додаткові інструменти для аналізу та візуалізації даних.

5. Висока доступність і надійність: завдяки можливості розгортати мікросервіси на різних серверах або контейнерах можна забезпечити високу доступність і надійність системи. Якщо один компонент виходить із ладу, інші можуть продовжувати працювати без перерви.

6. Скорочення часу налаштування та розгортання: Docker контейнери дають змогу однаково налаштовувати середовища розробки й виробництва, що спрощує розгортання й тестування нових версій сервісів.

7. Моніторинг і логування: використання Docker контейнерів у сполученні з мікросервісною архітектурою дає змогу легко налаштовувати моніторинг і збирання логів для кожного окремого компонента Data Pipeline, що допомагає вчасно виявляти й вирішувати проблеми.

Локальне рішення ELT Data Pipeline наведено на рис. 3. Користувач подає запит на моніторинг посту деякої соціальної медіа через HTTP до скрепера. Тут пропонується скрепер, написаний мовою Python з використанням FastAPI

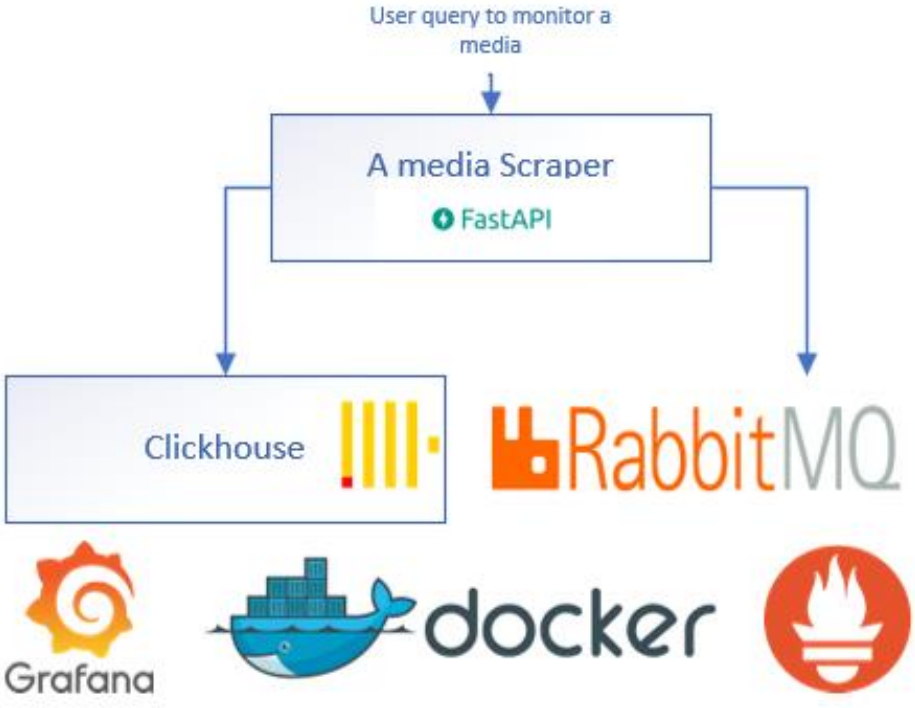

Рис. 3. Локальне рішення ELT Data Pipeline

фреймворку, що дає можливість оптимізувати швидкодію клієнт-серверного рішення порівняно з іншими фреймворками. Як правило, API соціальних медіа повертають дані порціями. Отримані дані надсилаються до брокера повідомлень, щоб покласти їх у чергу для подальшої обробки. Це може бути будь-який сервіс для роботи з чергами, чи AWS SQS, Kafka, чи RabbitMQ, для подальшої обробки в рамках ETL Data Pipeline (див. рис. 4). Отримані результати зберігаються до СД Clickhouse. Для логування та моніторингу роботи ETL Data Pipeline використовується Prometheus і Grafana.

Реалізація ETL Data Pipeline складається з вхідної черги RabbitMQ з повідомленнями, отриманими на попередньому етапі, консюмеру, який запускає відповідні мутації по різних потоках відповідно до схеми, наведеної на рис. 4, і вихідної черги RabbitMQ, у яку зберігаються результати обробки. Так організовується паралельна обробка даних незалежних скриптів і послідовне виконання залежних. Із вихідної черги окремий воркер збирає дані до відповідної структури (див. рис. 2) і зберігає до сховища.

Для оркестрації контейнерів пропонується використовувати Kubernetes, що дає змогу автоматизувати розгортання, масштабування й керування контейнерами, а також допомагає забезпечити надійну і стабільну роботу Data Pipeline й високу доступність системи.

Особливістю модулів обробки даних є використання Dask фреймворку. Dask дає змогу автоматично розподіляти обробку даних на багато вузлів або обчислювальних ресурсів і виконувати їх паралельно. Це прискорює обробку великих обсягів даних і пришвидшує обчислення, дає можливість масштабувати обробку даних зі зростанням їх обсягів, а також автоматично розподіляти завдання й динамічно керувати ресурсами, що допомагає уникнути перенавантаження та забезпечити ефективне використання обчислювальних ресурсів.

Деталі обробки даних у кожному модулі наведено на рис. 5–12. Кожен модуль відповідає за обробку даних на певному етапі роботи Data Pipeline та розрахунку необхідних складників розширеної думки [6].

Щодо використаних підходів, то розрахунки деяких модулів являють собою набір простих функцій (рис. 5–6), тоді як інші потребують використання специфічних бібліотек і сторонніх API для розрахунків. Наприклад, для розрахунку кількості орфографічних помилок можна використати бібліотеку language_tool_python (рис. 7) [14], а для визначення кількості пунктуаційних помилок – neuro_comma [15] і punctuator [16] (рис. 8).

Деякі модулі потребують інших підходів, як, наприклад, ті, що наведені на рис. 9–10. Для визначення факту написання власних імен із великої літери та кількості використаних нелітературних слів використовують словарний підхід. Варто зазначити, що словники для кожної задачі та мови різні. Словниковий підхід має низку

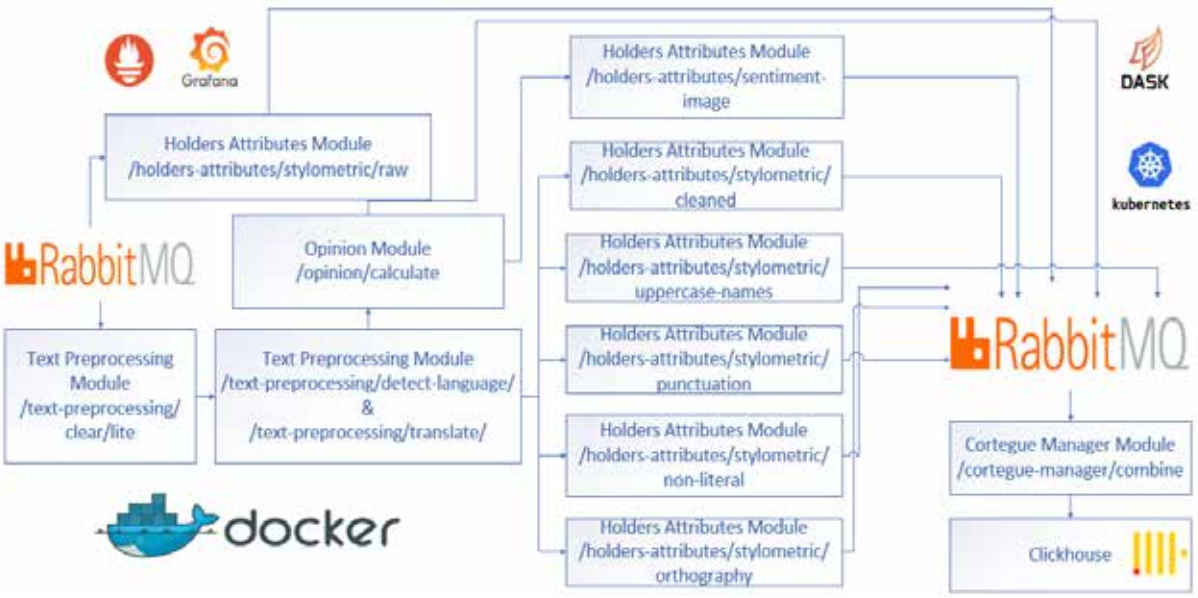

Рис. 4. Локальне рішення ETL Data Pipeline

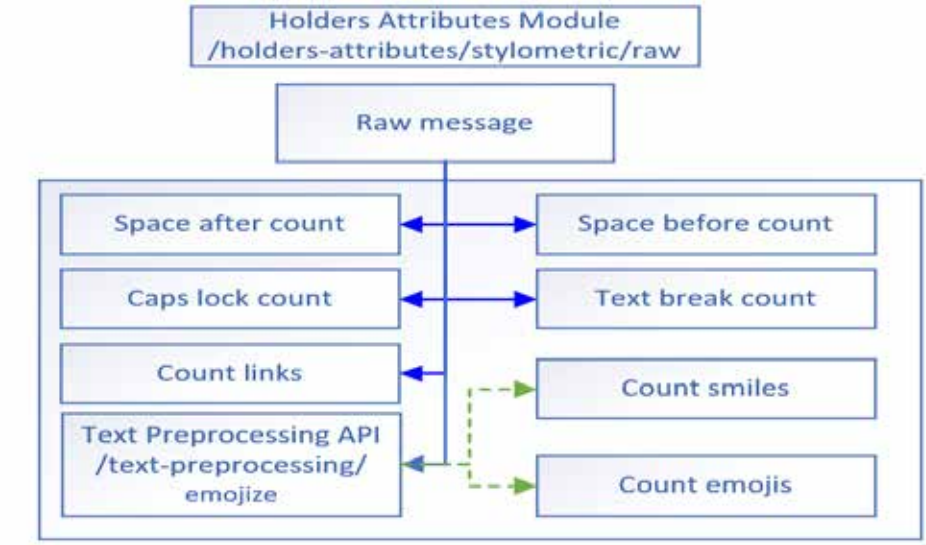

Рис. 5. Модуль обробки сирих даних для визначення деяких стилістичних атрибутів авторів думок

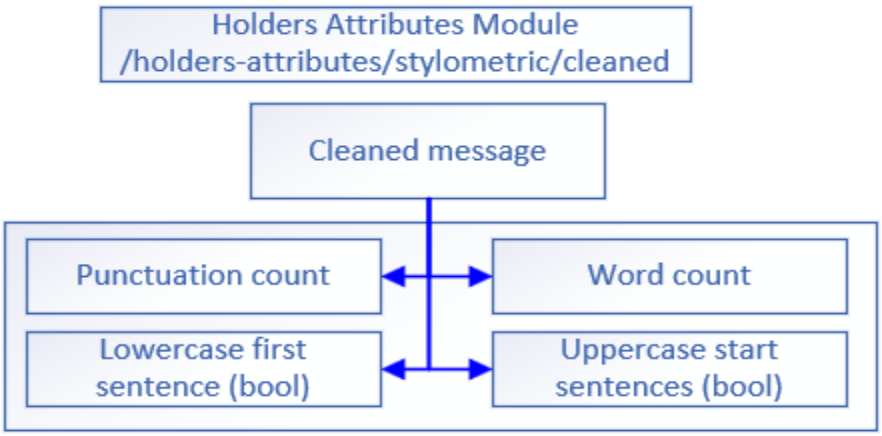

Рис. 6. Модуль обробки очищених даних для визначення деяких стилістичних атрибутів авторів думок

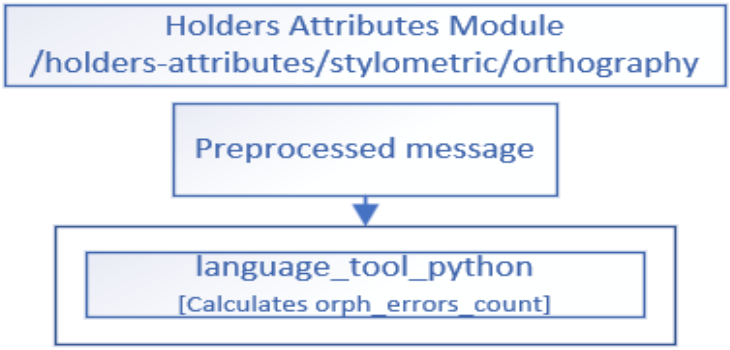

Рис. 7. Модуль обробки меседжу для визначення кількості орфографічних помилок

недоліків, як-то: оновлення словників новими даними та правилами, підтримка їх актуальності, важкість у роботі зі складними граматичними структурами, залежність від експертів. Словниковий підхід зазвичай ґрунтується на заздалегідь визначених словниках і правилах, тому він може бути менш ефективним у роботі з новими словами, що нечасто зустрічаються, або варіантами виразів.

Структуру модуля розрахунку сентимент-компонент складного текстового контенту соціальних медіа наведено на рис. 11. Розрахунок сентимент-

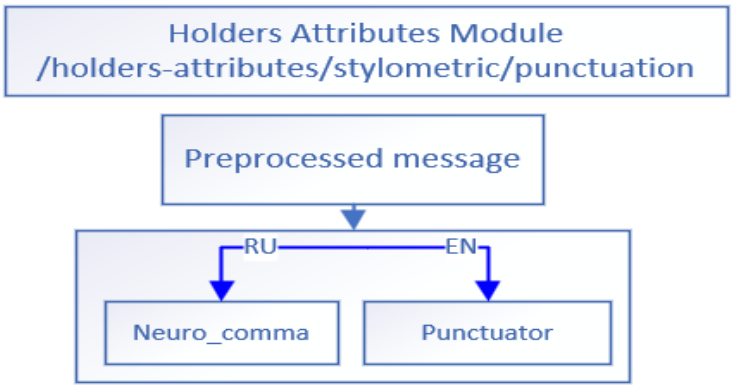

Рис. 8. Модуль обробки меседжу для визначення кількості пунктуаційних помилок

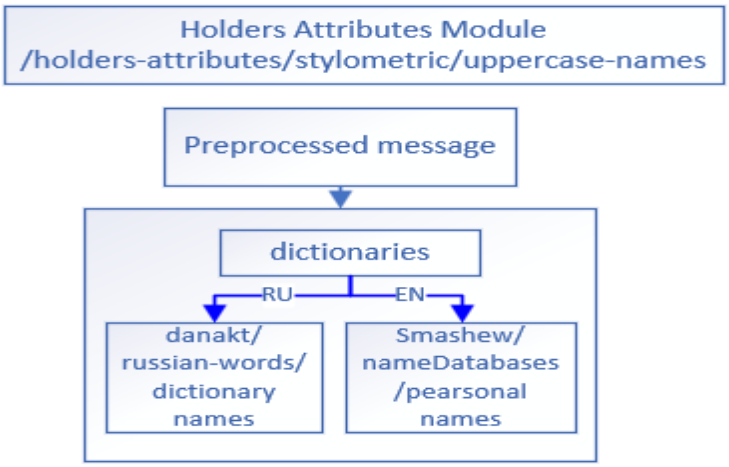

Рис. 9. Модуль обробки меседжу для визначення факту написання власних імен із великої літери

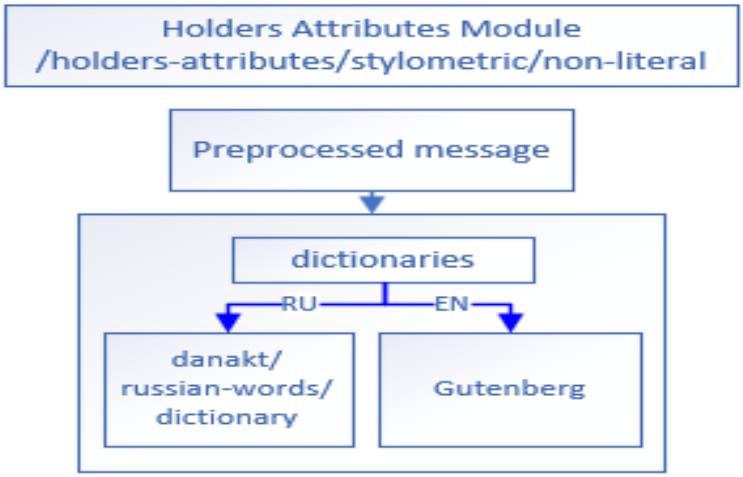

Рис. 10. Модуль обробки меседжу для визначення кількості використаних нелітературних слів

компонент пропонується реалізувати за допомогою таких бібліотек, як Polyglot, Dostoevsky, emosent, і використання регулярних виразів і словників для розрахунку складників на базі смайлів.

Модуль розрахунку сентимент-атрибутів авторів меседжів складного текстового контенту соціальних медіа (рис. 12) базується на сентимент-компонентах, отриманих на попередньому кроці (рис. 11) і являє собою здебільшого прості перевірки наявності компонент у складі кортежу. Оцінювання OP integral score є більш комплексним і складним в обчисленні [6].

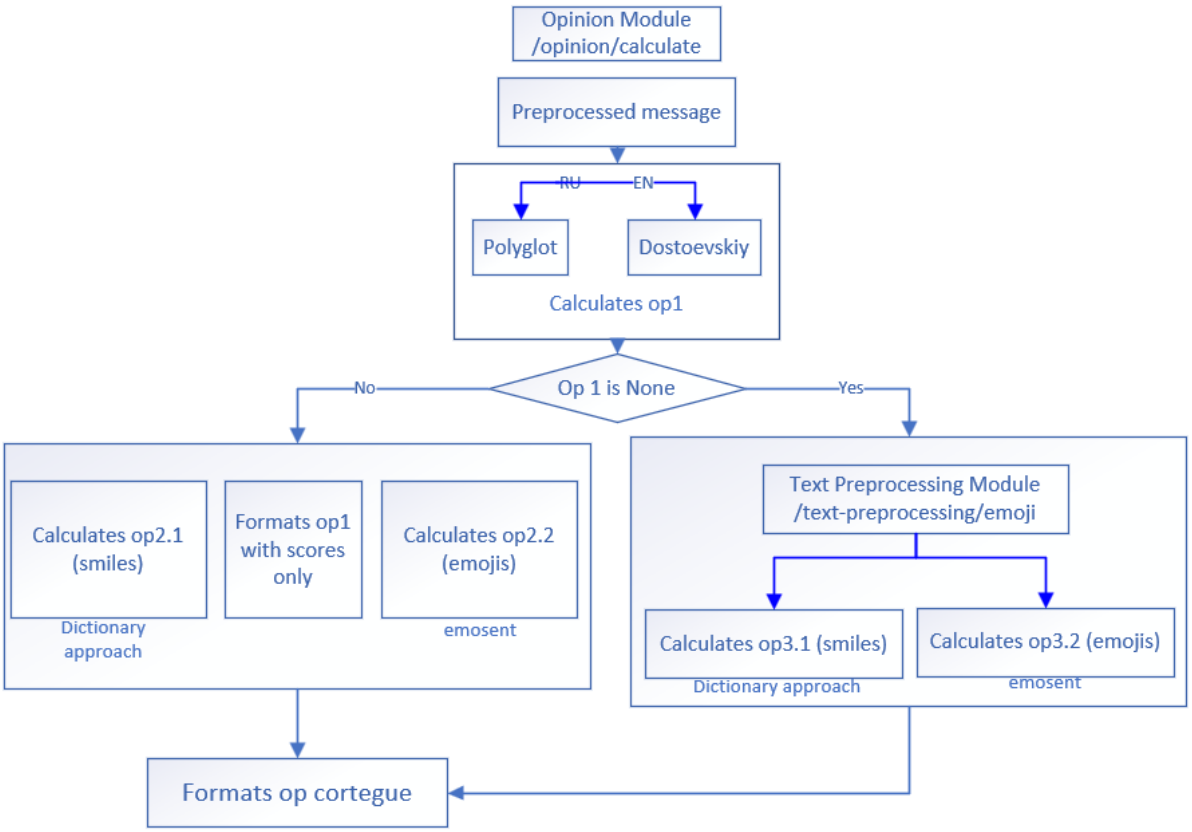

Рис. 11. Модуль розрахунку сентимент-компонент складного текстового контенту соціальних медіа

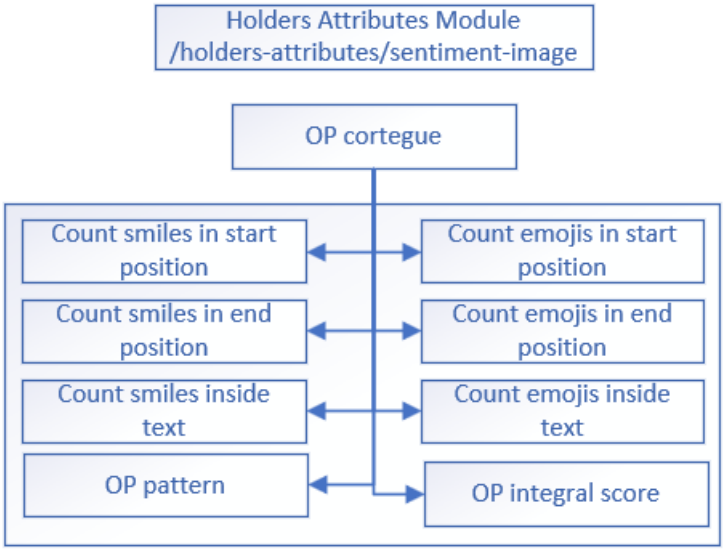

Рис. 12. Модуль розрахунку сентимент-атрибутів авторів меседжів складного текстового контенту соціальних медіа

Щодо хмарного підходу, то, на відміну від наявних рішень із використанням лямбда функцій [5], Amazon Kinesis зі зберіганням сирих даних до S3 [6] і частково хмарним рішенням на базі веб-сокетів [6], загальна схема локального рішення залишається незмінною, крім деяких деталей. ELT та ETL Data Pipelines у хмарі AWS (рис. 13–14) використовують Amazon EKS для оркестрації контейнерів, що дає змогу значно спростити перехід до іншої хмари в разі необхідності, аніж, наприклад, із використанням спеціалізованого Amazon ECS.

Рис. 13. ELT Data Pipeline в хмарі AWS

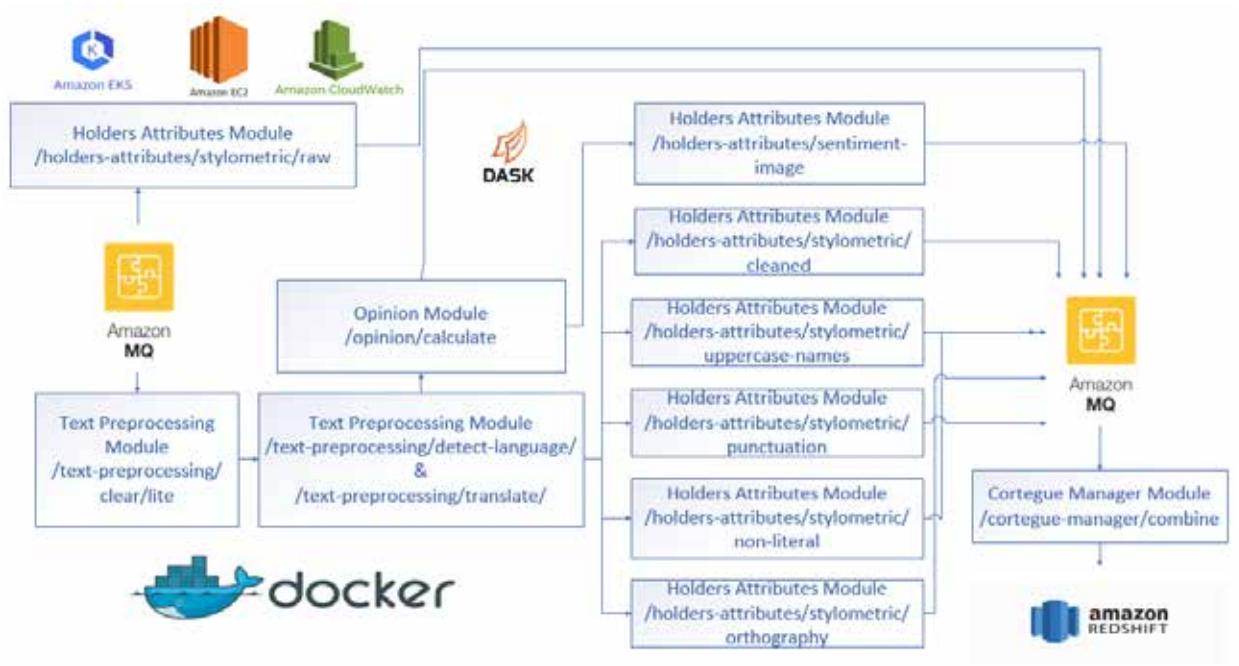

Рис. 14. ETL Data Pipeline даних у хмарі AWS

Як сховище даних використовують Amazon Redshift, переваги якого вже зазначені вище.

Для роботи з чергами пропонується використати Kafka (AWS MSK) або RabbitMQ (AWS MQ). У роботі обрано AWS MQ, серед вигід якого варто відзначити його характер як сервісу в хмарі Amazon.

Воркери побудовано на базі Amazon EC2 інстансів для обробки даних у режимі реального часу з використанням Dask фреймворку. Це дає змогу отримати більше гнучкості для обчислень і масштабування ресурсів порівняно з AWS Lambda.

Для логування та моніторингу використовується Amazon CloudWatch.

Модулі розрахунків, наведені на рис. 14, ті самі, що й у локальній схемі, і відповідають наведеним на рис. 4. Основною характеристикою розробленого рішення є його гнучкість та адаптивність. Можна розвернути локальне рішення з використанням ресурсів хмарних інтернет-провайдерів, чи в Amazon, чи в Google Cloud, чи в Microsoft Azure. Також має місце й комбінований підхід із використанням особливостей хмарних сервісів, який запропоновано в роботі як хмарне рішення.

Висновки. У роботі розглянуто питання архітектури хмарного й локального рішення інженерії даних у задачах моніторингу складного текстового контенту із соціальних медіа. Запропоновано два підходи, які базуються на концепції data mesh, для створення системи зберігання та обробки даних із можливістю адаптації до різних медіаплатформ.

Показано, що в контексті моніторингу соціальних медіа важливо використовувати потоковий ETL для швидкої обробки й аналізу даних, особливо враховуючи тенденцію до швидкого затухання дискусій.

Для локального рішення запропоновано використовувати ClickHouse завдяки його колонковій архітектурі, горизонтальному масштабуванню й ефективності в роботі з часовими рядами сентимент-оцінок. Для хмарного рішення обрано Amazon Redshift, який відзначається високою продуктивністю й інтеграцією з іншими сервісами AWS. Важливою частиною обох підходів є спільна схема сховища, основана на архітектурі зірки. Фактична таблиця, вимірювальні таблиці та ключі зв'язку визначають структуру, яка сприяє ефективному аналізу й злиттю даних.

Сховища для зберігання оброблених даних організовано також відповідно до принципів data mesh і складаються з денормалізованих таблиць, що сприяє прискоренню обробки й аналізу даних у режимі реального часу. Використання полів типу JSON у таблицях дає змогу зберігати вкладені структури даних без жорсткої прив'язки до структури таблиці. Це забезпечує гнучкість у додаванні нових атрибутів або оновленні наявних без необхідності зміни структури.

Використання Amazon EKS дає змогу ефективно керувати контейнерами й оркеструвати їх, що спрощує перехід до іншої хмарної інфраструктури в разі потреби.

Обробники, побудовані на базі Amazon EC2 для обробки даних у режимі реального часу, використовують Dask фреймворк для автоматичного розподілу обробки даних та ефективного використання ресурсів.

Запропоновані в роботі архітектурні рішення, на відміну від відомих, дають можливість створення гнучкої та ефективної системи моніторингу складного текстового контенту із соціальних медіа (з урахуванням емоджі, смайлів тощо), яка здатна пристосовуватися до змін у різних медіаплатформах і забезпечувати швидкий аналіз і відгук на актуальні події.

ЛІТЕРАТУРА

1. Modern Unified Data Architecture. URL: https://towardsdatascience.com/modern-unified-data-architecture-38182304afcc.

2. What is ETL? The Ultimate Guide: ETL Benchmark and Market Analysis. URL: https://www.castordoc.com/blog/etl-benchmark-for-mid-market-companies.

3. Data Lake Architecture: A Comprehensive Guide. URL: https://www.virtasant.com/blog/data-lake-architecture.

4. Data Mesh – Rethinking Enterprise Data Architecture. URL: https://www.cuelogic.com/blog/data-mesh.

5. Valeriy Sydorenko, Svitlana Kravchenko, Yurii Rychok, Karel Zeman. *Method of Classification of Tonal Estimations Time Series in Problems of Intellectual Analysis of Text Content* / Transportation Research Procedia. 2020. С. 102–109.

6. Heorhii Chyzhmak, Valeriy Sydorenko. *Classification models of direct opinion holders in the space of stylometric and sentiment features of chat messages* / MEES – 2023 IEEE 5th International Conference on Modern Electrical and Energy System.

7. Rychok, Y., Kravchenko, S., Sydorenko, V. (2018). A direct opinion model based on in-depth semantic and sentiment analysis of unstructured content in monitoring tasks of social networks. Proceedings of the XXV International scientific conference of young scientists and researchers «Topical problems of vital functions of society», 24–25 April (pp. 45–46). Kremenchuk, Ukraine: Kremenchuk Mykhailo Ostrohradskyi National University.

8. Liu, Bing (2010). Sentiment Analysis and Subjectivity. *Handbook of Natural Language Processing (Second ed.)*. Editors : In Indurkhya ; N. ; Damerau, F.J.

9. Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5 (4), 1093–1113. https://doi.org/10.1016/J.ASEJ.2014.04.011.

10. Wang, H., & Castanon, J.A. (n.d.). Sentiment Expression via Emoticons on Social Media.

11. Novak, P.K., Smailović, J., Sluban, B., & Mozetič, I. (2015). Sentiment of Emojis. *PLOS ONE*, 10 (12), e0144296. https://doi.org/10.1371/JOURNAL.PONE.0144296.

12. Ayvaz, S., & Shiha, M. O. (2017). The Effects of Emoji in Sentiment Analysis. *International Journal of Computer and Electrical Engineering*, 9 (1), 360–369. https://doi.org/10.17706/IJCEE.2017.9.1.360-369.

13. Novak, P.K., Smailović, J., Sluban, B., & Mozetič, I. (2015). Sentiment of Emojis. *PLOS ONE*, 10 (12), e0144296. https://doi.org/10.1371/JOURNAL.PONE.0144296.

14. language-tool-python PyPI. URL: https://pypi.org/project/language-tool-python.

15. Neuro-comma GitHub. URL: https://github.com/sviperm/neuro-comma.

16. punctuator PyPI. URL: https://pypi.org/project/punctuator.

CLOUD SOLUTION ARCHITECTURE AND DATA ENGINEERING ISSUES IN MONITORING COMPLICATED TEXTUAL CONTENT FROM SOCIAL MEDIA

Heorhii Chyzhmak
Postgraduate Student at the Department of Computer Engineering and Electronics
Kremenchuk Mykhailo Ostrohradskyi National University, 20 University str., Kremenchuk, Poltava region, Ukraine, 39600, george.chizhmak@gmail.com
ORCID: 0000-0001-9284-4195

Valerii Sydorenko
Doctor of Philosophy, Associate Professor at the Department of Computer Engineering and Electronics
Kremenchuk Mykhailo Ostrohradskyi National University, 20 University str., Kremenchuk, Poltava region, Ukraine, 39600, vnsidorenko@gmail.com
ORCID: 0000-0002-4449-073X

Vitalii Morvanyuk
Senior DevOps Engineer
AirSlate Poland, 12 M. Curie-Sklodowska str., Wroclaw, Poland 50-381, vitalii.morvaniuk@airslate.com
ORCID: 0009-0009-7816-5159

Tetiana Oleksienko
Master's degree at the Department of Computer Engineering and Electronics
Kremenchuk Mykhailo Ostrohradskyi National University, 20 University str., Kremenchuk, Poltava region, Ukraine, 39600, tanuyshka2002@gmail.com
ORCID: 0009-0002-0038-350X

The current stage of development of information technology is leading to a significant transformation in the management and analysis of large volumes of textual content originating from social media. Due to the large number of users and the redundancy of information generated on a daily basis, social media has become a powerful source of important data that can be used to understand global trends, analyze public opinion and make informed decisions in various fields.

A retrospective look at the history of the Internet reveals at least two obvious facts: a massive shift from slow communication (email, forum channels) to faster communication (chats and messengers) and a significant transformation of the language of communication itself. The authors do not aim to provide a comprehensive analysis of the reasons for this phenomenon, but consider it necessary to dwell on its features. The specifics of chatting lead to the fact that a holder tries to express his or her opinion briefly or very briefly, often using not only words but also various symbols, emojis, etc. On the other hand, modern communication is characterised by the use of slang, disregard for spelling, punctuation, and often moral standards. These features require the creation of appropriate architectural solutions for the efficient processing of this kind of data.

At the same time, the increase in the amount of information leads to challenges in monitoring and analyzing this textual content. The use of cloud and on-premises data engineering solutions to monitor complex textual content from social media is the key topic of this research.

The paper proposes storage and data processing architecture concepts that ensure scalability, reliability, and performance. At the same time, the proposed architectural solutions, unlike the known ones, make it possible to create a flexible and efficient system for monitoring complex textual content from social media (taking into account emojis, emoticons, etc.), which is able to adapt to changes in various media platforms and provide quick analysis and response to current events.

Key words: cloud solution architecture, data engineering, social media, monitoring, complex textual content, data mesh, streaming ETL, ETL, ELT, ClickHouse, Amazon Redshift, cloud technologies, data warehouse, Amazon EKS, Amazon EC2, Dask, RabbitMQ, AWS MQ, Data Pipeline.

REFERENCES

1. Modern Unified Data Architecture. Retrieved from https://towardsdatascience.com/modern-unified-data-architecture-38182304afcc.

2. What is ETL? The Ultimate Guide: ETL Benchmark and Market Analysis. Retrieved from https://www.castordoc.com/blog/etl-benchmark-for-mid-market-companies.

3. Data Lake Architecture: A Comprehensive Guide. Retrieved from https://www.virtasant.com/blog/data-lake-architecture.

4. Data Mesh – Rethinking Enterprise Data Architecture. Retrieved from https://www.cuelogic.com/blog/data-mesh.

5. Valeriy Sydorenko, Svitlana Kravchenko, Yurii Rychok, Karel Zeman. // Method of Classification of Tonal Estimations Time Series in Problems of Intellectual Analysis of Text Content / Transportation Research Procedia – 2020, с. 102–109.

6. Heorhii Chyzhmak, Valeriy Sydorenko. // Classification models of direct opinion holders in the space of stylometric and sentiment features of chat messages. / MEES – 2023 IEEE 5th International Conference on Modern Electrical and Energy System.

7. Rychok, Y., Kravchenko, S., Sydorenko, V. (2018). A direct opinion model based on in-depth semantic and sentiment analysis of unstructured content in monitoring tasks of social networks. Proceedings of the XXV International scientific conference of young scientists and researchers «Topical problems of vital functions of society», 24–25 April (pp. 45–46). Kremenchuk, Ukraine: Kremenchuk Mykhailo Ostrohradskyi National University.

8. Liu, Bing (2010). Sentiment Analysis and Subjectivity. *Handbook of Natural Language Processing (Second ed.)*. Editors: In Indurkhya, N.; Damerau, F.J.

9. Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4), 1093–1113. Retrieved from https://doi.org/10.1016/J.ASEJ.2014.04.011.

10. Wang, H., & Castanon, J. A. (n.d.). Sentiment Expression via Emoticons on Social Media.

11. Novak, P.K., Smailović, J., Sluban, B., & Mozetič, I. (2015). Sentiment of Emojis. *PLOS ONE*, 10(12), e0144296. Retrieved from https://doi.org/10.1371/JOURNAL.PONE.0144296.

12. Ayvaz, S., & Shiha, M.O. (2017). The Effects of Emoji in Sentiment Analysis. *International Journal of Computer and Electrical Engineering*, 9(1), 360–369. Retrieved from https://doi.org/10.17706/IJCEE.2017.9.1.360-369.

13. Novak, P.K., Smailović, J., Sluban, B., & Mozetič, I. (2015). Sentiment of Emojis. *PLOS ONE*, 10(12), e0144296. Retrieved from https://doi.org/10.1371/JOURNAL.PONE.0144296.

14. language-tool-python PyPI. Retrieved from https://pypi.org/project/language-tool-python.

15. Neuro-comma GitHub. Retrieved from https://github.com/sviperm/neuro-comma.

16. punctuator PyPI. Retrieved from https://pypi.org/project/punctuator.

Стаття надійшла 29.09.2023