УДК 004.89

АНАЛІЗ ТЕХНОЛОГІЙ ЗБИРАННЯ ІНФОРМАЦІЇ ДЛЯ АГРЕГАЦІЇ ШІ-ЧАТІВ

Вичалковський Д.А.[1]; Морозова Г.С.[2]
[1,2] Національний університет кораблебудування імені адмірала Макарова
[1,2] Україна, Миколаїв
[1] 231125@nuos.edu.ua; [2] ganna.morozova@nuos.edu.ua

Анотація. *Проаналізовано технології збирання даних для агрегації історії діалогів із кількох ШІ-чатів і акаунтів. Розглянуто три форми реалізації системи - застосунок, розширення браузера та вебсайт - і порівняно їх за критеріями автентифікації, доступу до даних, безпеки, кросплатформеності та складності розробки. Показано, що найпрактичнішою формою є застосунок з ізольованими сесіями акаунтів і єдиним пошуковим індексом, враховано обмеження умов використання провайдерів.*
Ключові слова: *ШІ-чати; агрегація даних; збирання інформації; застосунок; розширення браузера; вебсайт.*

Вступна частина. Користувачі дедалі частіше працюють одночасно з кількома ШІ-чатами (Claude, ChatGPT, Gemini, GLM тощо), часто з кількох акаунтів. Кожен сервіс зберігає власну історію діалогів у власному інтерфейсі, тому виникає фрагментація знань: користувач не пам'ятає, в якому чаті, під яким акаунтом і в якій нейромережі вирішувалася та чи інша задача. Жоден провайдер не зацікавлений у наскрізному пошуку по чатах конкурентів, тому ця ніша потребує незалежного сервісу-агрегатора.

Метою дослідження є визначення оптимальної технології та форми реалізації збирання даних із ШІ-чатів для сервісу-агрегатора.

Основна частина. Доступ до даних ШІ-чатів можливий кількома способами. Офіційні API стабільні, але надають лише дані, створені через самі API, а не історію веб-інтерфейсу. Офіційний експорт (наприклад, у ChatGPT або Google Takeout для Gemini) повний, але ручний і не працює в реальному часі. Внутрішні HTTP-запити та парсинг DOM дають повні й актуальні дані, проте залежать від змін верстки та API й можуть суперечити умовам використання сервісів. Автоматизація браузера (Playwright, Puppeteer) універсальна, але ресурсомістка й вимагає зберігання авторизації на сервері. Набір доступних технологій визначається формою реалізації системи, тому її вибір є центральним питанням.

Застосунок (Electron, Tauri) дозволяє створити окрему ізольовану сесію для кожного акаунта, зберігати авторизації в захищеному сховищі ОС, виконувати фонову синхронізацію та підтримувати локальний індекс. Його недоліки: потреба в установленні й оновленнях, збірки під Windows, macOS і Linux із підписом коду, обмеження мобільних платформ (лише WebView) і вищі витрати на розробку.

Розширення Chrome працює в уже авторизованому браузері, тому не потребує зберігання паролів і токенів сторонніх сервісів, має прямий доступ до сторінок чатів через content scripts, встановлюється в один клік і працює в Chromium-браузерах. Його обмеження - один профіль браузера означає одну сесію на домен (складна мультиакаунтність), обмеження Manifest V3, залежність від верстки, модерація Chrome Web Store та слабка підтримка мобільних пристроїв.

Вебсайт не потребує встановлення, доступний з будь-якого пристрою й зручний як єдине місце для інтерфейсу та індексу. Проте через політику Same-Origin і SameSite-cookies він не може напряму читати чати інших доменів: дані можна отримати лише через імпорт експортів,

API-ключі або передачу облікових даних на сервер, що ризиковано й часто порушує правила провайдерів. Централізоване зберігання чутливих діалогів також накладає відповідальність за захист даних.

Серед обмежень слід виділити правові: умови використання провайдерів можуть обмежувати автоматизований доступ до чатів [1–3], а дані користувачів підпадають під вимоги GDPR [4]. Технічні ризики пов'язані зі змінами API та верстки, тому потрібні модульні адаптери для кожного сервісу. Діалоги з ШІ часто містять чутливу інформацію, тому доцільні принцип local-first і шифрування даних.

Висновки. Кожна з форм має обмеження, але для поставленої задачі найпрактичнішим видається застосунок: він дозволяє підключати кожен ШІ-чат і кожен акаунт окремо з власною ізольованою авторизацією та зводити діалоги в єдиний індекс із загальним пошуком, при цьому він не залежить від профілю браузера й не потребує обходу обмежень браузерної безпеки.

Схема роботи:

- Користувач додає акаунт і входить у нього всередині застосунку
- Застосунок збирає історію чатів із кожної сесії окремим адаптером для кожного сервісу
- Діалоги приводяться до єдиного формату й потрапляють у спільний індекс.

Напрями розвитку:

- Автоматичний вибір нейромережі за типом задачі
- Повнотекстовий і семантичний пошук
- Теги й проєкти
- Дедуплікація діалогів.

Можливо в майбутньому:

Авторизація акаунтів через розширення браузера «в два кліки», що потребує окремого дослідження безпечної передачі сесії, обмежень Manifest V3 і політик провайдерів.

Висновок потребує підтвердження прототипом і перевірки актуальних умов використання та можливостей експорту кожного провайдера.

Використання інструментів ШІ. Під час підготовки тез використано ШІ-асистента Claude (Anthropic) для структурування матеріалу, пошуку та аналізу умов використання провайдерів і редагування тексту. Постановку задачі, вибір підходу та висновки визначав автор. Результати перевірено й відредаговано тим ж автором.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Anthropic. (б. д.). *Consumer Terms of Service*. https://anthropic.com/legal/terms
[2] OpenAI. (б. д.). *Terms of Use*. https://openai.com/terms/
[3] Google. (б. д.). *Terms of Service*. https://policies.google.com/terms
[4] EU. (2016). *Regulation 2016/679 (GDPR)*. https://eur-lex.europa.eu/eli/reg/2016/679/oj

Vychalkovskiy D.A., Morozova H.S.

Analysis of information collection technologies for aggregating AI chats

Abstract. *Three forms of an AI chat aggregator - a desktop application, a browser extension and a website - are compared by authentication, data access, security and development complexity. A desktop application with isolated account sessions and a unified search index is shown to be the most practical option given the providers' terms of use.*

Keywords: *AI chats; data aggregation; information collection; desktop application; browser extension; website.*