УДК 004.9

Євгеній КРИЖАНОВСЬКИЙ
Владислав ПОБІДАШ

ІНФОРМАЦІЙНА СИСТЕМА АНАЛІЗУ ВІДГУКІВ НА ПРОГРАМНІ ЗАСТОСУНКИ

Вінницький національний технічний університет

Анотація

Створено систему аналізу відгуків на програмні застосунки: збір, аналіз та класифікація відгуків для визначення та виправлення проблем на основі прямих відгуків від користувачів. Використовуючи сучасні методики аналізу даних, система може значно полегшити та покращити процес розробки програмного забезпечення.

Ключові слова: веб-додаток, інформаційна система, бази даних, аналіз даних, обробка природних мов, класифікація даних, інженерія підказок.

Abstract

A system for analyzing feedback on software applications has been created: collecting, analyzing, and classifying feedback to identify and fix problems based on direct feedback from users. Using modern data analysis techniques, the system can significantly facilitate and improve the software development process.

Keywords: web application, information system, databases, data analysis, natural language processing, data classification, prompt engineering.

Актуальність дослідження

У сучасному світі велика частина споживачів базує свій вибір програмного забезпечення на відгуках і рецензіях. Розуміння цього факту та здатність ефективно аналізувати та використовувати цю інформацію може значно вплинути на успіх розробки та маркетингу програмних продуктів. Інформаційна система, яка автоматизує збір, аналіз та візуалізацію відгуків, може надати цінний інструмент для зрозуміння потреб користувачів, виявлення слабких місць у програмному забезпеченні та вдосконалення продуктів з урахуванням фідбеку від користувачів.

Створення системи аналізу відгуків на програмні застосунки

Інформаційна система представляє собою веб-додаток, який розроблено на базі платформи Node.js згідно з full-stack архітектурою (рис. 1).

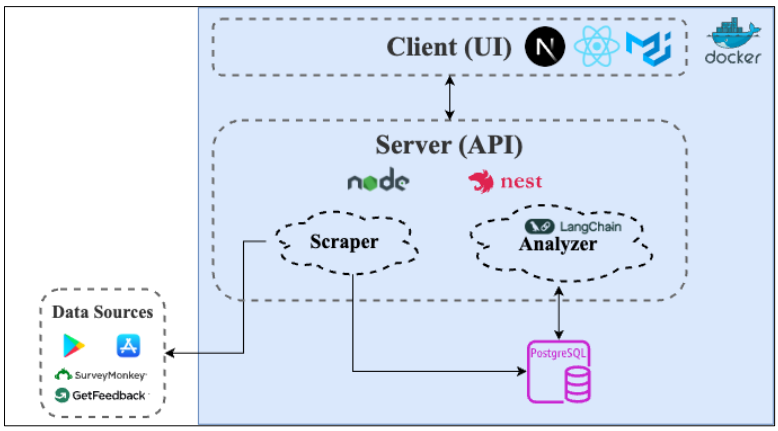

Рисунок 1 – Архітектура інформаційної системи аналізу відгуків

Front-end і back-end взаємодіють за допомогою REST API через безпечний протокол HTTPS. Контейнеризація за допомогою Docker гарантує ізоляцію, переносимість і ефективність розробки додатку на локальному рівні. Застосування потужної і надійної бази даних PostgreSQL дозволяє обробляти великі об'єми даних та складні запити. В системі вона використовується для зберігання відгуків та результатів їх аналізу.

Front-end створено на основі фреймворка Next.js, основою якого є JavaScript-бібліотека React, що використовується для розробки користувацьких інтерфейсів. Для back-end використано Nest.js – це сучасний Node.js фреймворк для створення ефективних, надійних і масштабних серверних додатків.

Модуль "scraper" відповідає за збір даних, зокрема відгуків користувачів. Основними джерелами для отримання цих відгуків є маркетплейси Google Play та App Store, де публікують додатки. Завдяки використанню таких Node.js бібліотек, як google-play-scraper та app-store-scraper цей процес значно спрощується (рис. 2). Також цей модуль можна масштабувати. Наприклад, зібрати відгуки для різних країн чи маркетплейсів і провести порівняльний аналіз для виявлення специфічних проблем конкретного ринку.

```
13  async scrapReviews({ appStoreUrl }: ScrapAppDto) {
14    const reviews = await this.scrapAppStoreReviews(appStoreUrl);
15
16    for (const { text, updated, score } of reviews) {
17      await this.reviewsService.create({
18        source: Source.APP_STORE,
19        text,
20        score,
21        timestamp: new Date(updated),
22      });
23    }
24  }
25
26  private async scrapAppStoreReviews(url: string) {
27    return await store.reviews({
28      id: this.extractId(url, /id(\d+)/),
29      country: 'ua',
30    });
31  }
```

Рисунок 2 – Приклад збору відгуків на App Store

Збір відгуків також може здійснюватися через сторонні сервіси, такі як GetFeedback чи SurveyMonkey. Будь-яка платформа, яка має текстові відгуки користувачів, може бути джерелом даних. Тому головну ідею даної інформаційної системи може бути легко застосовано для аналізу будь-яких відгуків, будь-то локації в Google Maps чи товари на Amazon.

Для застосування класичних методів аналізу даних потрібно мати гарне розуміння машинного навчання, обробки природного мовлення (NLP), статистики та лінгвістики. Також, необхідними є навички програмування на Python – основною мовою для обробки даних та машинного навчання. Вона має безліч бібліотек для обробки даних, статистики, машинного навчання та NLP, включаючи Pandas, NumPy, SciPy, Scikit-Learn, NLTK, TextBlob, SpaCy, Gensim та інші. Нижче наведено перелік основних методів, за допомогою яких виконується аналіз тексту.

- Сентимент-аналіз: Можна використовувати бібліотеки NLP, такі як TextBlob або NLTK як основу для моделі машинного навчання, яка класифікує відгуки на позитивні, негативні та нейтральні.
- Тематичне моделювання: Дозволяє виявити основні теми у відгуках, для цього можуть використовуватися такі методики, як метод головних компонент (PCA) або латентний розподіл Діріхле (LDA), що доступний через такі бібліотеки як Gensim.

- Аналіз контенту: Полягає в оцінці того, кількість елементів конкретного вмісту присутні в тексті. Бібліотеки NLP, такі як NLTK або Spacy, можна використати для токенізації та стемінгу для отримання "сирих" слів і фраз для аналізу
- Ізоляція ключових слів: Це спосіб знаходження найбільш значущих слів в тексті. Зазвичай це слова, які найчастіше зустрічаються або які найбільш показові для аналізованої теми. Можна використовувати алгоритми як TF-IDF (частота терміну - обернена частота документів) для відбору ключових слів в тексті.
- Машинне навчання: Часто використовуються в аналізі текстів алгоритми машинного навчання, такі як поверхневе навчання (supervised learning) та навчання без вчителя (unsupervised learning), що включають класифікацію тексту, кластеризацію та ін. Бібліотеки як Scikit-learn можуть бути використані для навчання моделей на відміченому датасеті.
- Методики обробки природних мов (NLP): Для більш глибокого аналізу можуть використовуватися бібліотеки NLP, як Spacy або NLTK. Зокрема, дані бібліотеки можуть використовуватися для POS-тегування, синтаксичного аналізу відношень тощо.

Виконання подібних аналізів може бути важким і потребувати значних обчислювальних ресурсів, особливо при роботі з великими наборами даних. Для цього, може бути необхідним використання високопродуктивних комп'ютерів або обчислювальні хмарні ресурси, особливо для тренування складних моделей машинного навчання.

В ході розробки інформаційної системи було прийнято рішення піти шляхом використання попередньо натренованих моделей. Такий підхід відомий як навчання з перенесенням (transfer learning) – це метод машинного навчання (ML), в якому знання, отримані при виконанні завдання, повторно використовуються для підвищення продуктивності при виконанні пов'язаного завдання [1]. Це дозволяє економити значний час та обчислювальні ресурси, які були б необхідні для тренування масштабної моделі із нуля.

Однак попередньо натреновані моделі можуть не давати високу точність при роботі з конкретними даними або в специфічному контексті. В таких випадках точність залежить від якості формулювання запиту та наявності відповідних підказок. Підказка – це інструкція природною мовою, яка вказує як виконати завдання. Інженерія підказок - це процес створення та вдосконалення підказок, які використовує модель [2]. Модель слідує за підказкою, щоб визначити структуру та зміст тексту, який їй потрібно проаналізувати чи згенерувати.

LangChain – це фреймворк, який використовує існуючі мовні моделі з контекстом (підказками, прикладами, обґрунтуванням відповіді та іншими джерелами) для формування відповідей та визначення дій на основі наданого контексту [3-5]. Для створення об’єкту моделі достатньо лише кілька стрічок коду (рис. 3).

```
13  private chat(options?: Partial<{ temperature: number; modelName: string }>) {
14    return new ChatOpenAI({
15      temperature: options.temperature ?? 0,
16      modelName: options.modelName ?? 'gpt-3.5-turbo',
17    });
18  }
```

Рисунок 3 – Метод для створення чату OpenAI на базі моделі GPT

Далі можна гнучко описати запит, використовуючи відповідні шаблони, які надає LangChain та за допомогою метода pipe створити послідовність виконуваних модулів, передаючи вихідні дані одного модуля в інший модуль або подібний до нього. Таким чином, було імплементовано метод для перекладу відгуків (рис. 4).

```
async translate(text: string) {
  const prompt = ChatPromptTemplate.fromMessages([
    [
      'system',
      `
        You are a world class translator.
        Your answer is just a translation.
      `,
    ],
    [
      'human',
      `
        Translate text into English.
        If text is already in English, just return it without changes.
        Text: {text}
      `,
    ],
  ]);

  const chain = prompt
    .pipe(this.chat({ temperature: 0 }))
    .pipe(new StringOutputParser());

  try {
    const translation = await chain.invoke({ text });

    return translation;
  } catch (error) {
    console.error('Something went wrong during translation.', error);

    return '';
  }
}
```

Рисунок 4 – Переклад тексту з використанням методів LangChain

MetadataTagger – це ще один зручний метод, який надає бібліотека LangChain. Він використовує конфігурований ланцюжок на основі OpenAI Functions та виконує генерацію метаданих з кожного наданого документа відповідно до заданої схеми (рис. 5).

```
async tag(text: string) {
  const zodSchema = z.object({
    mood: z.enum(['delighted', 'happy', 'sad', 'angry', 'neutral']),
    insights: z.array(
      z.object({
        phrase: z.string(),
        tone: z.enum(['positive', 'neutral', 'negative']),
        category: z
          .string()
          .describe('propose an app feature depends on a phrase'),
      }),
    ),
  });

  const metadataTagger = createMetadataTaggerFromZod(zodSchema, {
    llm: this.chat({ temperature: 0.15 }),
    prompt: PromptTemplate.fromTemplate(`
      You received a feedback for a streaming application.

      Do the following:
      1. identify unique by meaning phrase
      2. identify tone of each phrase
      3. identify general mood of a reviewer

      Remove: simple communication phrases; just phrase
      Output: JSON format only

      Review: {input}
    `),
  });

  try {
    const taggedDocument = await metadataTagger._transformDocument(
      new Document({ pageContent: text }),
    );

    return taggedDocument.metadata as { mood: Mood; insights: Insight[] };
  } catch (error) {
    console.error('Something went wrong during tagging', error);
  }
}
}
```

Рисунок 5 – Аналіз даних з використанням методів LangChain

В результаті було зібрано відгуки користувача, перекладено їх текст, визначено настрій та ключові фрази (рис. 6).


```
{
  id: 'e5ad3b8c-8ae6-4d76-bc2a-1e2cbfc7a4de',
  source: 'APP_STORE',
  text: '1. Відсутній контент на українській мові\n2. Мало контента на руском языке',
  translation: '1. There is no content in Ukrainian language\n' +
    '2. There is little content in Russian language',
  score: 3,
  timestamp: 2020-03-17T17:02:40.000Z,
  mood: 'neutral',
  insights: [
    {
      phrase: 'There is no content in Ukrainian language',
      tone: 'negative',
      category: 'language support'
    },
    {
      phrase: 'There is little content in Russian language',
      tone: 'neutral',
      category: 'language support'
    }
  ]
}
```


Рисунок 6 – Приклад опрацьованого відгуку користувача

На їх основі можна побудувати графіки та зробити відповідні висновки щодо ключових проблем конкретного застосунку (рис. 7).

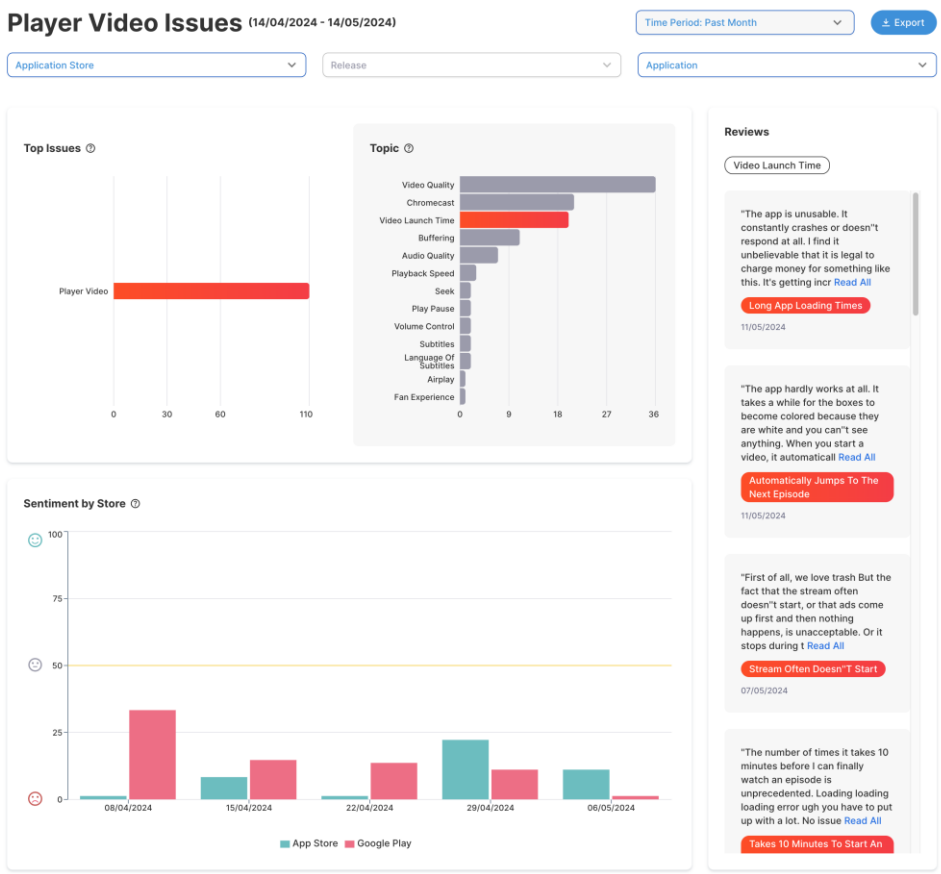

Рисунок 7 – Інтерфейс користувача для перегляду результатів аналізу відгуків на застосунок

Висновки

Отже, для виконання стандартного аналізу даних не обов’язково створювати власні моделі, а цілком достатньо скористатись існуючими та зосередитись на формулюванні запитів та наданні необхідного контексту. LangChain – чудовий фреймворк, який дозволяє все це зробити. Розроблена інформаційна система підтверджує ефективність такої стратегії

та допомагає в прийнятті рішень для розвитку бізнесу, використовуючи потенціал штучного інтелекту.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Spring Research Presentation: A Theoretical Foundation for Inductive Transfer. – West, Jeremy; Ventura, Dan; Warnick, Sean – Brigham Young University, College of Physical and Mathematical Sciences, 2007
2. Overview of prompts: [Електронний ресурс] – URL: https://learn.microsoft.com/en-us/ai-builder/prompts-overview
3. LangChain: [Електронний ресурс] – URL: https://js.langchain.com/docs/get_started/introduction
4. Наука про дані: машинне навчання та інтелектуальний аналіз даних : електронний навчальний посібник комбінованого (локального та мережевого) використання [Електронний ресурс] / В. Б. Мокін, М. В. Дратований – Вінниця : ВНТУ, 2024. – 258 с.
5. Мокін В. Б., Бондалєтов К. О., Крижановський Є. М., і Караваєв В. О. Метод аугментації текстів про стан масивів вод на основі інтелектуальної прив'язки до багатозв'язних геоінформаційних систем іменованих сутностей, Вісник Вінницького політехнічного інституту, вип. 3, с. 55–65, Черв. 2023. URL: https://doi.org/10.31649/1997-9266-2023-168-3-55-65

Євгеній Миколайович Крижановський – канд. техн. наук, доцент кафедри системного аналізу та інформаційних технологій, Вінницький національний технічний університет, Вінниця, e-mail: kruzhan@gmail.com;

Побідаш Владислав Віталійович – студент групи 2ІСТ-20б, Факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, Вінниця, e-mail: vladpobidash@gmail.com;

Evgeniy Kryzhanovsky M. – Cand. Sc. (Eng), Department of Systems Analysis and Information Technology, Vinnytsia National Technical University, Vinnytsia, e-mail: kruzhan@gmail.com;

Vladislav Pobidash V. - student of 2IST-20b group, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: vladpobidash@gmail.com.