УДК 004.8

**О.О. Шинкаренко**
**О.В. Сілагін**

# РОЗРОБКА ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ АНАЛІЗУ ТЕКСТІВ НА НАЯВНІСТЬ ОБРАЗЛИВИХ ВИСЛОВЛЮВАНЬ

Вінницький національний технічний університет

***Анотація***

*Досліджується проблема створення інформаційної технології із веб-інтерфейсом та механізмом доступу за API-ключами для аналізу текстів на наявність образливих, неприйнятних висловів, нецензурної лексики за декількома критеріями, яка допоможе пришвидшити процес модерації коментарів, повідомлень користувачів у соціальних мережах, форумах, комп'ютерних іграх тощо. Також дана система допоможе акцентувати увагу робітників-модераторів на розгляд більш спірних випадків вживання образливих висловлювань.*

**Ключові слова:** веб-інтерфейс, образливі висловлювання, нецензурна лексика, модерація, API-ключ.

***Abstract***

*The problem of creating information technology with a web interface and access mechanism based on API-keys for analyzing texts for the presence of offensive, unacceptable utterances, obscene language on several criteria is being explored which will help speed up the process of moderation of comments, user messages on social networks, forums, computer games, etc. This system will also help to focus the attention of moderators on the consideration of more controversial cases of usage of abusive language.*

**Keywords:** web interface, offensive language, obscene language, moderation, API-key.

## Вступ

На сьогоднішній більшість онлайн-ресурсів, таких як платформи розповсюдження новин, платформи відповідей на запитання та обміну досвідом, соціальні мережі, форуми, а також онлайн-ігри включають в себе простір для спілкування, висловлення думок. З ціллю збереження поважного ставлення до співрозмовника та плідного обговорення тем, компанії наймають контент-модераторів, які слідкують за тим, аби правила обговорення платформи виконувались. У разі невиконання таких правил, модератори видаляють дописи користувачів та, за необхідності, блокують і самого користувача. Відповідно, модератори повинні правильно класифікувати та визначити причину видалення коментаря та повідомити про неї автора. Така робота є рутинною та вимагає великих часових затрат на прийняття рішень, а зі збільшенням кількості коментарів для розгляду виникає потреба у наймі більшої кількості працівників[1, 2]. Процес можливо пришвидшити та спростити, якщо відсортовувати коментарі по ступеню ймовірності порушення правил спілкування платформи автоматизовано та пропонувати на розгляд модераторів у пріоритеті більш спірні випадки порушення правил.

## Постановка задачі

На ринку існують розроблені рішення даної проблеми. Першим прикладом є система BattlEye, що спеціалізується на модерації спілкування в комп'ютерних іграх та слідкує за використанням нелегального програмного забезпечення з ціллю отримання переваги у грі. Система працює у реальному часі та є автоматизованою. Головним недоліком даної системи є автоматичне блокування підозрілих дій та сумнівних коментарів, тому для оскарження такого рішення необхідно писати звернення у службу підтримки, де кожен з випадків розглядається окремо. Іншим прикладом є Ethical AI, що спеціалізується на створенні штучного інтелекту для модерації під потреби замовника. Головним недоліком даної системи є необхідність розробки нового рішення під кожен індивідуальний випадок. Також аналогом рішення, що розглядається, є Perspective API – програмний інтерфейс аналізу текстів за встановленими критеріями. Даний інтерфейс легко інтегрувати в існуючі програмні рішення та використовувати отримані дані для власних цілей. Головним недоліком даної системи є відсутність

власного функціоналу і візуалізації даних, отриманих в результаті аналізу, тому у випадках подальшої обробки отриманої інформації необхідно додатково реалізовувати власний функціонал, що несе за собою додаткові грошові та часові витрати [3, 4]..

Усі вищенаведені комерційні рішення у тій чи іншій формі стикаються із проблемою масштабування. Із масштабуванням програмного забезпечення на більшу кількість користувачів також виникає проблема забезпечення децентралізації сховищ даних та доступу до них. Рішення «під ключ» із збільшенням кількості клієнтів зустрічають проблему росту необхідних ресурсів, як матеріальних, так і часових, на створення нового рішення. В той же час, універсальні рішення не надають індивідуальних можливостей при роботі із проаналізованими даними, в тому числі і взагалі не забезпечують зберігання та модерацію проаналізованого тексту. З цього виникає проблема децентралізованого і ізольованого доступу до даних, для вирішення якої необхідно враховувати функціональну повноту системи та економічну доцільність впровадження.

Тому стоїть проблема створення інформаційної технології із веб-інтерфейсом та механізмом децентралізованого та ізольованого доступу до даних клієнта для автоматизованого аналізу тексту на елементів, що порушують правила платформи, та пріоритетного пропонування на розгляд працівникам-модераторам на розгляд спірних, неоднозначних випадків порушення.

*Метою дослідження є* розширення функціональних можливостей в аналізі текстів на наявність образливих висловлювань. На відміну від аналогів, які не забезпечують механізму децентралізованого ізольованого доступу до даних, пропонується поєднання інтелектуального модуля із механізмом доступу до даних за API-ключами.

*Об'єктом дослідження є* процеси аналізу текстів на наявність образливих висловлювань, їх пріоритезації.

*Предметом дослідження є* алгоритми та програмне забезпечення, що організовує процес аналізу текстів на наявність образливих висловлювань, їх пріоритезацію та доступ до збережених даних за допомогою API-ключів.

## Результати дослідження

Класифікація – процес групування та організації інформацію змістовно та систематично у стандартному форматі, що використовується для виявлення схожості ідей, подій, об’єктів, осіб, явищ[10]. Особливістю класифікації є те, що групи, на які мають бути поділені дані(класи) заздалегідь відомі та описані, тобто представляють собою скінчену дискретну множину категорій, до яких можна віднести об’єкти аналізу. При перенесенні задачі класифікації на аналіз тексту, можна виділити декілька основних підходів:

- класифікація окремо взятих слів або висловлювань у тексті;
- класифікація синтаксично зв’язаного тексту.

В даному дослідженні розглядається класифікація синтаксично зв’язаного тексту

Друга група методів, а саме класифікація зв’язного тексту, включають декілька етапів. Першим етапом є перетворення вхідних даних у такі, що можуть використовуватись для подальшого аналізу нейронними мережами. У задачах аналізу тексту використовують векторне представлення слів(word embeddings). Методи векторного представлення слів дозволяють перетворити вхідний текст в малорозмірні вектори, що і називається векторним представленням[5,13]. Такий метод дозволяє представити дані у більш компактному вигляді та представити зв’язки слів між собою(рис. 1).

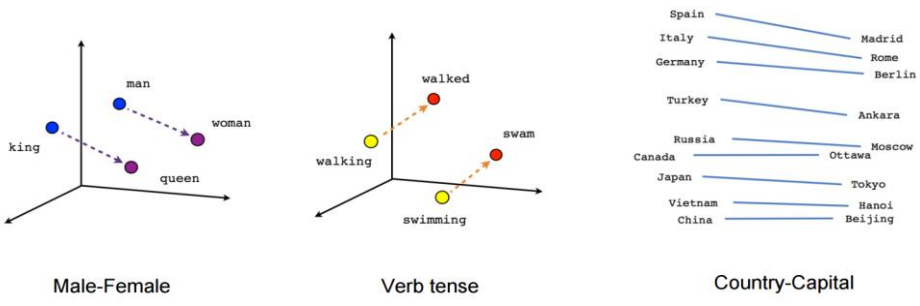


Рисунок 1 – Приклад векторного представлення слів.

Для векторного представлення слів використовуються різні моделі кодування. Найбільше для вирішення задачі підходять дві з них, а саме модель кодування речень на основі трансформатора(transformer based encoding) та модель кодування на основі глибокої усереднювальної мережі(Deep Averaging Network). Для проектування та реалізації було обрано універсальний кодувальник речень з векторним перетворенням речень на основі трансформатора для досягнення вищого показника точності, а також через необхідність аналізу синтаксично зв'язного тексту.

Загальна структурна схема інтелектуального модуля аналізу текстів на наявність образливих висловлювань складається з 5 основних компонентів:

- веб-клієнт;
- програмний інтерфейс додатку(API);
- універсальний кодувальник речень;
- веб-панель адміністрування;
- база даних.

Дана схема показана на рисунку 2. Схема не враховує роботу із модулем більше ніж одного користувача та не забезпечує механізм розподіленого ізольованого доступу до даних.

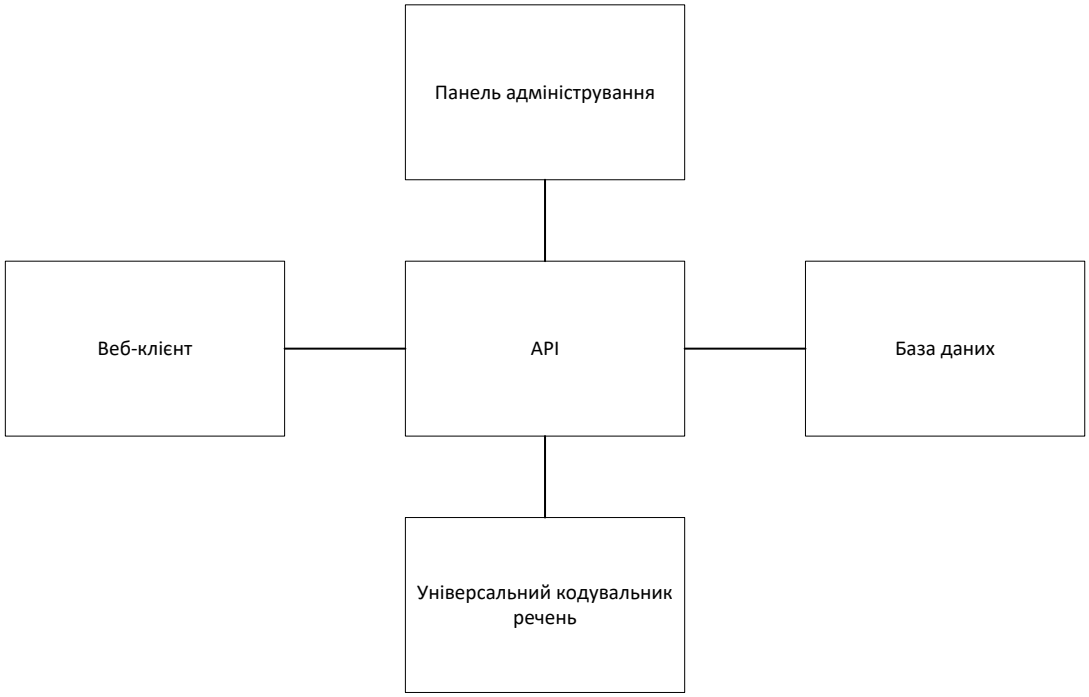


Рисунок 2 – загальна структурна схема інтелектуального модуля аналізу текстів на наявність образливих висловлювань

Таким чином, враховуючи вищезазначену проблему, модифікуємо структурну схему, додавши до неї 2 нових компоненти: модуль авторизації із прив'язкою API-ключа та маршрутизатор бази даних за API-ключем. Дані компоненти забезпечують єдиний шлюз доступу до даних за допомогою API-ключа та дозволяють розподіляти дані на різних структурних одиницях бази даних, спрощуючи процес пошуку даних. Цим також вирішується проблема масштабування та комерційного впровадження, робота нових користувачів із системою ніяк не впливатиме на сесію інших користувачів.

Крім того, для впровадження системи як набору засобів розробки, необхідно додати транслятор змін інформації про допис – коли користувач(чи сервер) формує запит на аналіз коментаря, ідентифікатор якого вже зберігається в базі даних, транслятор повинен надіслати відому йому інформацію про допис. Таким чином, можна впровадити фільтрацію дописів на стороні сервера-користувача системи аналізу текстів, значно спростивши інтеграцію механізму модерації у готові програмні продукти. Модифікована структурна схема системи аналізу текстів на наявність образливих висловлювань наведе на рисунку 3.

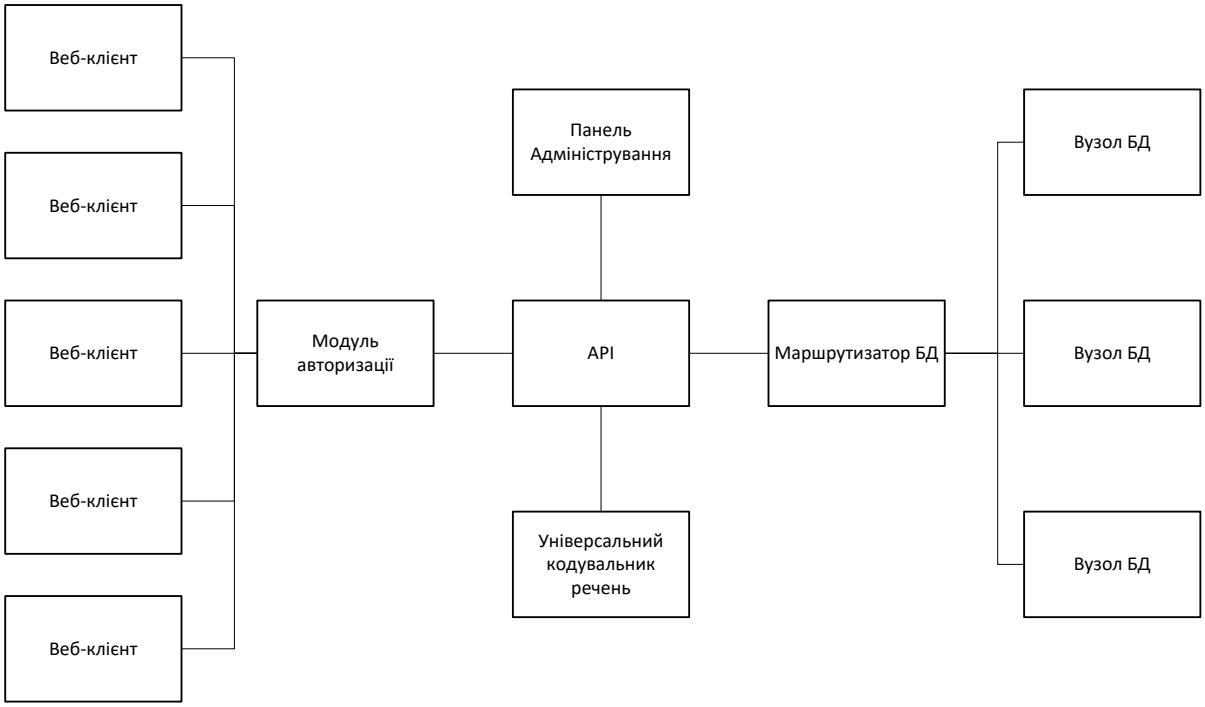


Рисунок 3 – загальна структурна схема системи інтелектуального аналізу текстів на наявність образливих висловлювань

Для отримання фінансової вигоди з впровадження інтелектуальної системи за API-ключами є можливим використання моделі розповсюдження наборів для розробки програмного забезпечення. Серед наявних варіантів: різні тарифні плани з обмеженням трафіку згідно плану; розповсюдження системи на безкоштовній основі з обмеженнями трафіку; розповсюдження інтелектуальної системи на безкоштовній основі із монетизацією через рекламні оголошення. Для визначення кращого підходу необхідно провести додаткові дослідження тенденцій та підходів на ринку.

## Висновки

1. Встановлено, що рішення для модерації текстів мають комерційний попит та вирішують актуальну проблему збереження раціонального та поважного ставлення до співрозмовників.
2. Використання штучного інтелекту для аналізу текстів значно зменшує рутинність праці модераторів та надає більше можливостей для індивідуального формування висновку щодо шкоди різних дописів і коментарів.
3. Існуючі рішення не забезпечують належний рівень свободи масштабування та не надають універсального механізму ізоляції даних окремого користувача, а також складно інтегруються в існуючі програмні продукти.
4. Підхід із використанням API-ключів вирішує проблему швидкого доступу до даних, зменшує навантаження на систему та на працеспроможність вже існуючих сесій інших користувачів.
5. Модуль транслятора дозволяє легко інтегрувати інтелектуальну систему для вирішення специфічних завдань індивідуально кожному клієнтові через систему запитів на сервер та відповіді у вигляді наявної у базі даних інформації про допис за ідентифікатором.
6. Питання монетизації(отримання прибутку) з дитрибуції інтелектуальної системи аналізу текстів потребує додаткового дослідження існуючих підходів ринку, зокрема, способи монетизації існуючих SDK(Software Development Kit).

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. 1. J. Risch, R. Ruff, R. Krestel. Offensive Language Detection Explained, 2020. — 137c.

2. 2. R. Pradhan, A. Chaturvedi, A. Tripathi, D.K. Sharma. A Review on Offensive Language Detection [Електронний ресурс] – Режим доступу: https://www.researchgate.net/publication/338355806_A_Review_on_Offensive_Language_Detection.
3. О. Шинкаренко. Розробка інтелектуальної системи аналізу тексту на наявність образливих висловлювань [Електронний ресурс] – режим доступу: https://conferences.vntu.edu.ua/index.php/mn/mn2021/paper/view/11738
4. О. Шинкаренко. Розробка додатку інтелектуального аналізу тексту на наявність образливих висловлювань [Електронний ресурс] – режим доступу: https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2021/paper/view/12600
5. E. Hoffmann. Standard Statistical Classifications: Basic Principles.
6. K. Pykes. Vector Space Models [Електронний ресурс] – Режим доступу: https://towardsdatascience.com/vector-space-models-48b42a15d86d.
7. T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean. Distributed Representations of Words and Phrases and their Compositionality
8. Why and when to use API keys [Електронний ресурс] – режим доступу: https://cloud.google.com/endpoints/docs/openapi/when-why-api-key

***Шинкаренко Олег Олександрович*** **–** студент кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця. e-mail: oshynkarenko1503@gmail.com .

***Сілагін Олексій Віталійович*** **–** канд. техн. наук, доцент кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця. e-mail: avsilagin@vntu.edu.ua .

***Shynkarenko Oleh Oleksandrovych*** **–** student of Computer Science Department, Vinnytsia National Technical University, Vinnytsia, e-mail: oshynkarenko1503@gmail.com.

***Silagin Oleksiy Vitalyevich*** **–** Cand Sc. (Eng.), Associate Professor of Computer Science Department, Vinnytsia National Technical University, Vinnytsia, e-mail: avsilagin@vntu.edu.ua.