МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.
_____Юрій КОНДРАТЕНКО
«____»_____2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА

ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ АНАЛІЗУ КОНТЕНТУ ТА ВИЯВЛЕННЯ ТЕНДЕНЦІЙ У СОЦІАЛЬНИХ МЕРЕЖАХ

Спеціальність 124 «Системний аналіз»
Освітня програма «Системний аналіз»

Здобувачка _____Катерина КОЛЄСНІЧЕНКО
«___»_____ 2024 р.

Керівник канд. фіз.-мат. наук, доцент _____ Інесса КУЛАКОВСЬКА
«___»_____ 2024 р.

Миколаїв – 2024

Чорноморський національний університет імені Петра Могили

(повне найменування закладу вищої освіти)

Факультет	Комп'ютерних наук
Кафедра	Інтелектуальних інформаційних систем
Рівень вищої освіти	Другий (магістерський)
Освітній ступень	Магістр
Спеціальність	Системний аналіз
Освітня програма	Системний аналіз

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних інформаційних систем

_____ Юрій КОНДРАТЕНКО

«____» _____2024 р.

ЗАВДАННЯ

на кваліфікаційну роботу здобувачки

Колєсніченко Катерини Олегівни

(прізвище, ім'я, по батькові здобувачки)

1. Тема кваліфікаційної роботи: «Інформаційна система для аналізу контенту та виявлення тенденцій у соціальних мережах».

Керівник роботи: Кулаковська Інесса Василівна, доцент кафедри ІІС, канд. фіз.-мат. наук, доцент.

Затверджена наказом ЧНУ ім. Петра Могили від «03» червня 2024 р. № 140/1.

2. Строк представлення кваліфікаційної роботи «17» грудня 2024 р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні: розробка інформаційної системи для автоматизованого аналізу контенту з соціальних мереж, виявлення негативних тенденцій та класифікація даних; вихідні дані – текстовий контент з соціальних мереж для аналізу.

4. Перелік питань, що підлягають розробці: огляд методів обробки текстових даних та автоматизованого аналізу контенту; застосування методів обробки природної мови (NLP) та аналізу настроїв для виявлення негативних тенденцій.

5. Перелік графічних матеріалів: КР – 100 сторінок, 48 рисунків, 1 таблиця, 40 джерел, 1 додаток та презентація.

Керівник роботи _____ Інесса КУЛАКОВСЬКА
(Особистий підпис) *(Власне ім'я ПРІЗВИЩЕ)*

Здобувачка _____ Катерина КОЛЄСНІЧЕНКО
(Особистий підпис) *(Власне ім'я ПРІЗВИЩЕ)*

Дата видачі завдання «07» червня 2024 р.

КАЛЕНДАРНИЙ ПЛАН
кваліфікаційної роботи

Тема: Інформаційна система для аналізу контенту та виявлення тенденцій у соціальних мережах

№	Найменування роботи	Початок	Закінчення	Примітки
1	Отримання завдання на виконання КР	03.06.2024	09.06.2024	Виконано
2	Аналіз предметної області та постановка задачі	11.06.2024	21.06.2024	Виконано
3	Огляд літературних джерел за темою кваліфікаційної роботи, зокрема аналіз з публікацій щодо обробки текстових даних і виявлення негативних тенденцій	21.06.2024	01.07.2024	Виконано
4	Огляд існуючих методів NLP для виявлення негативних тенденцій	01.09.2024	25.10.2024	Виконано
5	Розробка системи автоматизованого аналізу контенту, оцінка ефективності методів виявлення негативних тенденцій	26.10.2024	21.11.2024	Виконано
6	Перший попередній захист КР на засіданні комісії кафедри	22.11.2024	22.11.2024	Виконано
7	Корегування роботи за результатами попереднього захисту	23.11.2024	05.12.2024	Виконано
8	Другий попередній захист КР на засіданні комісії кафедри	06.12.2024	06.12.2024	Виконано
9	Доробка та остаточне оформлення КР	07.12.2024	10.12.2024	Виконано
10	Подання КР, її електронної копії та інших документів (відгуку, рецензії) до захисту	16.12.2024	17.12.2024	Виконано

Керівник роботи _____ Інесса КУЛАКОВСЬКА
(Особистий підпис) *(Власне ім'я ПРІЗВИЩЕ)*

Здобувачка _____ Катерина КОЛЄСНІЧЕНКО
(Особистий підпис) *(Власне ім'я ПРІЗВИЩЕ)*

Дата складання календарного плану
«21» червня 2024 р.

АНОТАЦІЯ

магістерської кваліфікаційної роботи студентки групи 607 ЧНУ ім. Петра Могили

Колєсніченко Катерини Олегівни

Тема: «Інформаційна система для аналізу контенту та виявлення тенденцій у соціальних мережах»

Актуальність роботи: в умовах постійного зростання популярності соціальних мереж виникають серйозні проблеми, такі як мова ненависті, кібербулінг і дезінформація, що створюють ризики для користувачів та підривають довіру до платформ. Ефективне виявлення цих негативних явищ є необхідним для забезпечення безпечного середовища в соціальних мережах.

Об'єкт роботи – процес аналізу контенту в соціальних мережах для виявлення негативних тенденцій.

Предмет роботи – інформаційна система, що розроблена для автоматизованого аналізу контенту з використанням методів обробки природної мови та аналізу настроїв.

Метою роботи є дослідження ефективних методів для виявлення негативних явищ, таких як мова ненависті, та розробка алгоритмів для їх автоматизованого виявлення.

Основний текст кваліфікаційної роботи складається зі вступу, чотирьох розділів, висновків та додатків.

У першому розділі розглядаються проблеми негативного контенту в соціальних мережах та методи їх виявлення.

У другому розділі розглядаються етичні та правові аспекти виявлення негативного контенту в соціальних мережах.

Третій розділ присвячений методам і моделям, які використовуються для автоматизованого аналізу текстових даних. В даному розділі детально описуються основні алгоритми для класифікації текстів, зокрема на основі методів машинного навчання та аналізу настроїв. Описано використання бібліотек та інструментів,

таких як VADER (для аналізу настроїв), PRAW (для збору даних з Reddit), а також інших платформ, які використовуються для автоматичного збору даних з соціальних мереж, їх аналізу та виявлення токсичних або ненавистних висловлювань. Також розглянуто можливості використання глибокого навчання (наприклад, моделей BERT) для точнішого розпізнавання емоційного тону та токсичних коментарів.

Четвертий розділ присвячений проєктуванню та реалізації інформаційної системи для автоматизованого аналізу контенту в соціальних мережах. Структура системи включає кілька основних компонентів: збір даних через API Reddit за допомогою бібліотеки PRAW, аналіз настроїв коментарів із застосуванням методів VADER, TextBlob та BERT, визначення hate speech за допомогою моделі Toxic BERT, а також фільтрація коментарів за настроєм і кількістю лайків для уточнення результатів. Для візуалізації даних використовується бібліотека matplotlib, що дозволяє виводити графіки розподілу настроїв у реальному часі. Система інтегрує ці компоненти в єдину платформу, що забезпечує автоматизований збір, обробку, аналіз та візуалізацію результатів, дозволяючи користувачам ефективно відслідковувати негативні тенденції в онлайнових обговореннях.

Магістерська кваліфікаційна робота містить 100 сторінок, 48 рисунків, 1 таблицю , 45 використаних джерел та 1 додаток.

Ключові слова: *аналіз контенту, соціальні мережі, мова ненависті, кібербулінг, дезінформація, класифікація настроїв, обробка природної мови, інформаційна система.*

ABSTRACT

To the master's qualification work of a student of group 607 at Petro Mohyla Black Sea National University

Koliesnichenko Kateryna

Topic: "Information system for content analysis and trend detection in social networks"

Relevance of work: with the continuous growth in the popularity of social networks, serious issues such as hate speech, cyberbullying, and disinformation arise, which create risks for users and undermine trust in platforms. The effective detection of these negative phenomena is crucial for ensuring a safe environment on social networks.

The object of the work is the process of content analysis in social networks for detecting negative trends.

The subject of the work is an information system designed for automated content analysis using natural language processing methods, sentiment analysis.

The purpose of the work is to investigate effective methods for detecting negative phenomena, such as hate speech, cyberbullying, and disinformation, and to develop algorithms for their automated detection.

The main text of the qualification work consists of an introduction, four chapters, conclusions, and appendices.

In the first chapter, the issues of negative content in social networks and methods for detecting such content are examined.

The second section examines the ethical and legal aspects of identifying negative content on social media.

The third chapter is devoted to methods and models used for automated text data analysis. This section describes in detail the main algorithms for text classification, including those based on machine learning and sentiment analysis. The chapter describes the use of libraries and tools such as VADER (for sentiment analysis), PRAW (for collecting data from Reddit), and other platforms used to automatically collect data from social media, analyze it, and identify toxic or hateful statements. The possibilities

of using deep learning (e.g., BERT models) to more accurately recognize emotional tone and toxic comments are also considered.

The fourth chapter is devoted to the design and implementation of an information system for automated content analysis on social media. The architecture of the system includes several main components: data collection via the Reddit API using the PRAW library, analysis of comment sentiment using VADER, TextBlob, and BERT methods, hate speech detection using the Toxic BERT model, and filtering comments by sentiment and number of likes to refine the results. For data visualization, the matplotlib library is used, which allows you to display graphs of sentiment distribution in real time. The system integrates these components into a single platform that provides automated collection, processing, analysis, and visualization of results, allowing users to effectively monitor negative trends in online discussions.

The master's qualification work contains 99 pages, 48 figures, 1 table, 45 references, and 1 appendix.

Keywords: *content analysis, social networks, hate speech, cyberbullying, disinformation, sentiment classification, natural language processing, information system.*

ЗМІСТ

ВСТУП

Сучасне інформаційне суспільство характеризується швидким розвитком цифрових технологій, які впливають на всі аспекти нашого життя. У статті [1] розглядається вплив соціальних мереж на суспільну комунікацію. Одним із найяскравіших прикладів цього впливу є соціальні мережі, що стали важливим інструментом спілкування для мільярдів людей по всьому світу. Завдяки соціальним мережам користувачі можуть підтримувати соціальні контакти, обмінюватися інформацією, дізнаватися новини, висловлювати свої думки, знаходити професійні можливості та навіть вирішувати різноманітні проблеми. Соціальні платформи надають миттєвий доступ до величезної кількості інформації, що значно полегшує спілкування та взаємодію між людьми.

Однак, поряд із перевагами, які надають соціальні мережі, вони також породжують і нові виклики. Як зазначено в [2], соціальні мережі стали ключовим джерелом інформації для мільйонів користувачів. Одним із найбільш поширених негативних явищ у соціальних медіа є поширення хейтспічу (мови ненависті), кібербулінгу та дезінформації. Ці явища можуть мати серйозні наслідки для користувачів і суспільства загалом, адже вони можуть спричиняти психологічні проблеми, руйнувати соціальні зв’язки, сприяти зростанню соціальної напруженості та навіть загрожувати демократичним процесам через маніпуляції громадською думкою. Зокрема, дезінформація може впливати на політичні вибори, соціальні рухи або призводити до поширення неправдивих чуток, що породжують паніку серед населення.

Існуючі системи модерації контенту часто не можуть впоратися з великими обсягами даних, які щодня генеруються на платформах. Зазвичай модерація здійснюється вручну або за допомогою примітивних фільтрів, що не завжди дозволяє вчасно виявляти і блокувати негативний контент. Саме тому розробка автоматизованих систем, здатних в режимі реального часу аналізувати інформацію

і виявляти потенційно шкідливі матеріали, є одним із ключових напрямків у сучасних інформаційних технологіях.

Метою даного дослідження є не тільки розробка інформаційної системи для автоматизованого аналізу контенту в соціальних мережах, але й дослідження існуючих методів і технологій у цій галузі. Зокрема, дослідження спрямоване на аналіз сучасних алгоритмів природної мовної обробки (NLP) та аналізу настроїв, які можуть бути використані для виявлення негативного контенту, такого як мова ненависті або образи.

1 АНАЛІЗ КОНТЕНТУ СОЦІАЛЬНИХ МЕРЕЖ

1.1 Характеристика негативних тенденцій у соціальних мережах

Соціальні мережі стали важливою частиною сучасного інформаційного простору, забезпечуючи зручний спосіб комунікації, обміну інформацією та вираження думок. Однак їхнє масове використання також породжує серйозні проблеми, зокрема негативні явища, що можуть завдати шкоди як окремим індивідуумам, так і суспільству в цілому.

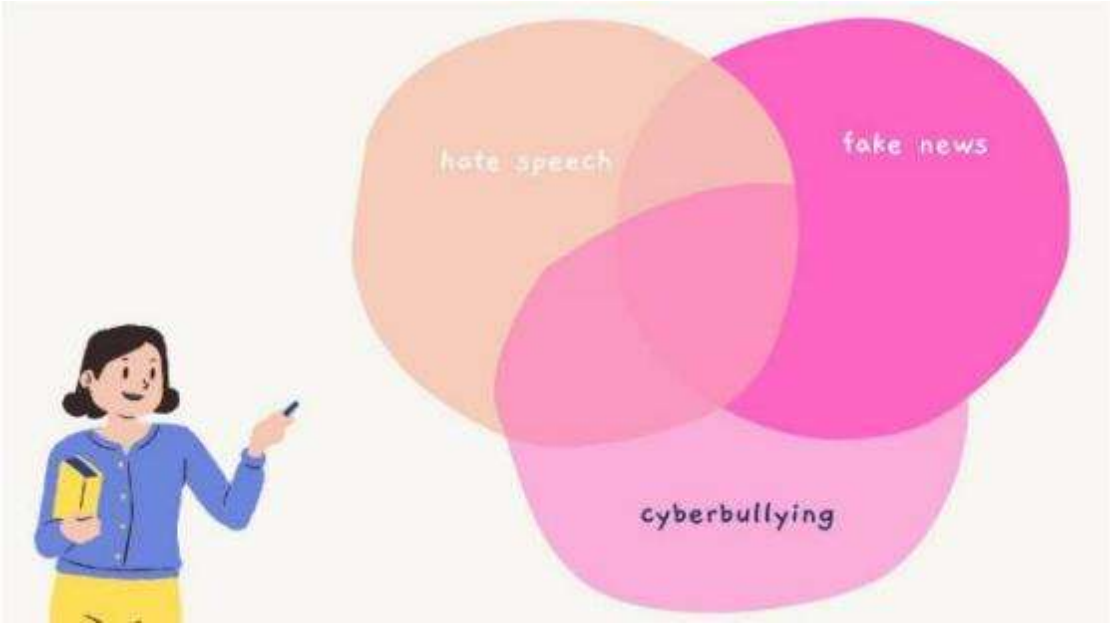

Рисунок 1.1 – Основні негативні тенденції

Основні негативні тенденції, які широко поширені в соціальних мережах, описані нижче.

У статті [3] доводиться, що мова ненависті є однією з найпоширеніших форм негативного контенту в соціальних мережах. Мова ненависті (hate speech) є однією з найбільш небезпечних форм негативного контенту в інтернеті. Вона зазвичай спрямована на певні групи людей, дискримінуючи їх за ознакою раси, релігії, етнічної приналежності, сексуальної орієнтації, гендеру або інших характеристик. Коментарі, що містять мову ненависті, зазвичай супроводжуються агресивними висловлюваннями, образами та неприязністю. Це може створювати ворожнечу між

різними групами населення, а також підвищувати рівень насильства в реальному житті.

Приклад мови ненависті: "Ці люди не мають права бути в нашій країні, вони тільки заважають. Треба вигнати всіх іноземців!" З точки зору системного аналізу, хейтспіч представляє складну задачу для автоматизованого виявлення через кілька факторів:

- неструктурованість даних. Хейтспіч може бути представлений у різних формах – текстових повідомленнях, мемах, відео чи аудіозаписах, що ускладнює його структурний аналіз;
- сарказм та двозначність. Виявлення хейтспічу стає складнішим через використання сарказму, метафор або жартів, що можуть бути неправильно інтерпретовані автоматизованими системами;
- різноманітність мовних конструкцій. Хейтспіч може бути виражений через набір нетипових мовних конструкцій, скорочень, сленгу або нецензурних виразів, які не завжди можуть бути розпізнані алгоритмами обробки природної мови (NLP).

Для виявлення хейтспічу використовуються методи обробки текстових даних, зокрема, обробка природної мови (NLP), кластеризація та класифікація на основі алгоритмів машинного навчання. Системний аналіз у цьому контексті включає розробку та оцінку ефективності алгоритмів для виявлення патернів у текстових даних, які асоціюються з хейтспічем.

Дослідження в [4] показує, що кібербулінг має серйозні наслідки для психічного здоров'я підлітків, зокрема у контексті соціальних медіа. Кібербулінг – це форма насильства, яка відбувається у мережі, найчастіше в соціальних мережах або інших інтернет-ресурсах. Це знущання, приниження, погрози або шантаж, які можуть мати серйозні наслідки для психічного здоров’я потерпілих. Кібербулінг може проявлятися у вигляді образливих коментарів, поширення принизливих чуток, створення фальшивих акаунтів для залякування або навіть відкритих погроз.Особливо важливою є проблема кібербулінгу серед молоді, адже він може

призвести до депресії, тривожних розладів або навіть самогубств у крайніх випадках.

Приклад кібербулінгу: "Ти ніколи не станеш популярним, не варто навіть намагатися. Ти просто жахливий, всі це знають."

Для виявлення такого контенту необхідні алгоритми, що можуть оцінити емоційну складову тексту та визначити рівень агресії чи насильства в коментарях. На відміну від традиційного булінгу, кібербулінг має кілька особливостей, що роблять його особливо небезпечним:

— анонімність: переслідувачі можуть залишатися анонімними, що знижує ймовірність відповідальності за свої дії;

— швидкість поширення: інформація в соціальних мережах поширюється швидко, що може призвести до масового залучення інших користувачів у процес переслідування;

— постійна доступність: жертви кібербулінгу можуть стикатися з переслідуванням 24/7, що призводить до психологічного тиску.

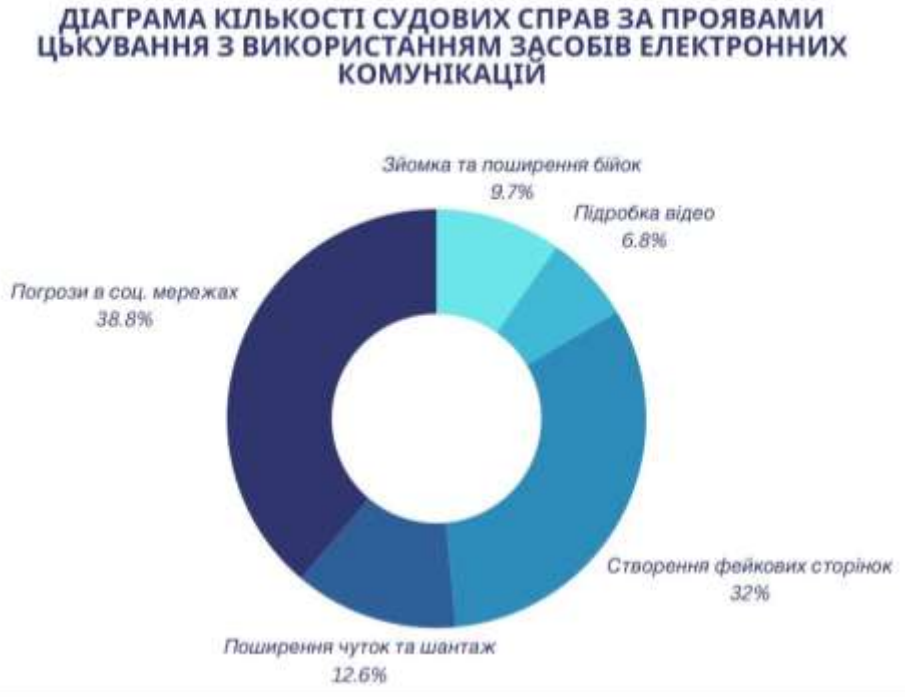

Рисунок 1.2 – Антирейтинг популярних видів кібербулінгу

Аналіз кібербулінгу в рамках системного аналізу охоплює дослідження моделей його поширення в мережі та розробку автоматизованих засобів виявлення. Зокрема, важливим є виявлення мереж користувачів, які взаємодіють з контентом, що містить кібербулінг, та аналіз їхньої активності.

Для виявлення кібербулінгу використовуються схожі до хейтспічу методи.

– класифікація тексту: застосовуються різні моделі класифікації для виявлення патернів агресивного або образливого мовлення;

– сентимент-аналіз: використовується для визначення загального тону повідомлень, що допомагає ідентифікувати потенційно небезпечний контент;

– соціальна мережа та аналіз графів: дослідження взаємодій між користувачами допомагає виявити групи або окремих осіб, що є ініціаторами кібербулінгу.

Дезінформація (fake news). В інтернеті активно поширюються неправдиві чи маніпулятивні новини, що можуть серйозно спотворювати реальну картину подій, викликати паніку або політичну нестабільність. Дезінформація в соціальних мережах – це навмисне поширення неправдивої або вводячої в оману інформації з метою маніпулювання думкою суспільства. Вона може мати значні наслідки для політичної стабільності, громадського порядку та здоров'я населення (наприклад, у випадку з фальшивими новинами про пандемію COVID-19). Як зазначено в [5], дезінформація в соціальних мережах може викликати політичну нестабільність і підривати довіру до владних структур.

У статті [6] обговорюється вплив фальшивих новин на громадську думку, що розповсюджується через соціальні мережі.

1.2 Методи аналізу текстового контенту

У сучасному світі велика кількість інформації генерується кожного дня, і в тому числі величезні обсяги текстових даних, що надходять через соціальні мережі. Для ефективного аналізу цього контенту та виявлення важливих патернів та тенденцій використовуються методи обробки природної мови (NLP) та аналіз

настроїв. У дослідженні [7] описано основні проблеми виявлення мови ненависті на онлайн-платформах, що ускладнює ефективне реагування на таку діяльність.

Обробка природної мови (Natural Language Processing, NLP) є галуззю штучного інтелекту, що займається взаємодією між комп'ютерами та людською мовою. Вона дозволяє машинам розуміти, інтерпретувати та генерувати людську мову таким чином, щоб зробити її корисною для аналізу та обробки.

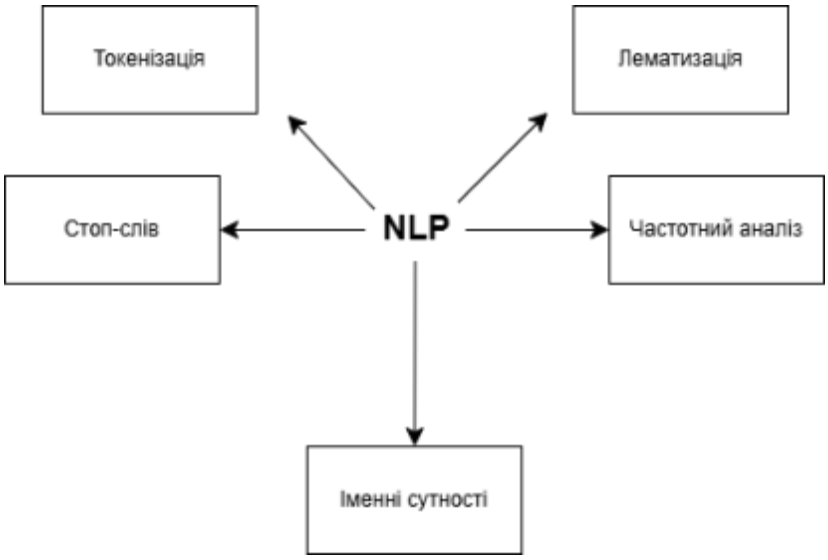

Рисунок 1.3 – Основні завдання NLP

Основні завдання NLP включають:

– токенізація: процес розбиття тексту на окремі одиниці, такі як слова, фрази або речення. Токенізація допомагає системі зрозуміти структуру тексту та аналізувати його частини;

– стоп-слів (stopwords): це загальні слова (наприклад, "і", "на", "в"), які часто не несуть суттєвої інформації для аналізу. Вони зазвичай видаляються на етапі попередньої обробки тексту;

– лематизація: це процес приведення слова до його початкової форми (леми). Наприклад, слово "ходив" буде лематизовано в "ходити". Це важливо для зменшення варіативності слів та полегшення їх аналізу;

– частотний аналіз: визначення найбільш вживаних слів або фраз у тексті для подальшого вивчення тенденцій;

– іменні сутності (Named Entity Recognition, NER): визначення імен, дат, місць або інших специфічних термінів у тексті для структурування даних.

Після виконання цих кроків можна отримати очищений та структурований текст, готовий до подальшого аналізу.

Аналіз настроїв є підгалуззю NLP, що займається класифікацією тексту за його емоційним забарвленням – позитивним, негативним або нейтральним. Основною метою аналізу настроїв є визначення емоційного тону коментаря, публікації чи іншого тексту. Аналіз настроїв має кілька основних етапів, котрі зображені на рис. 1.5.

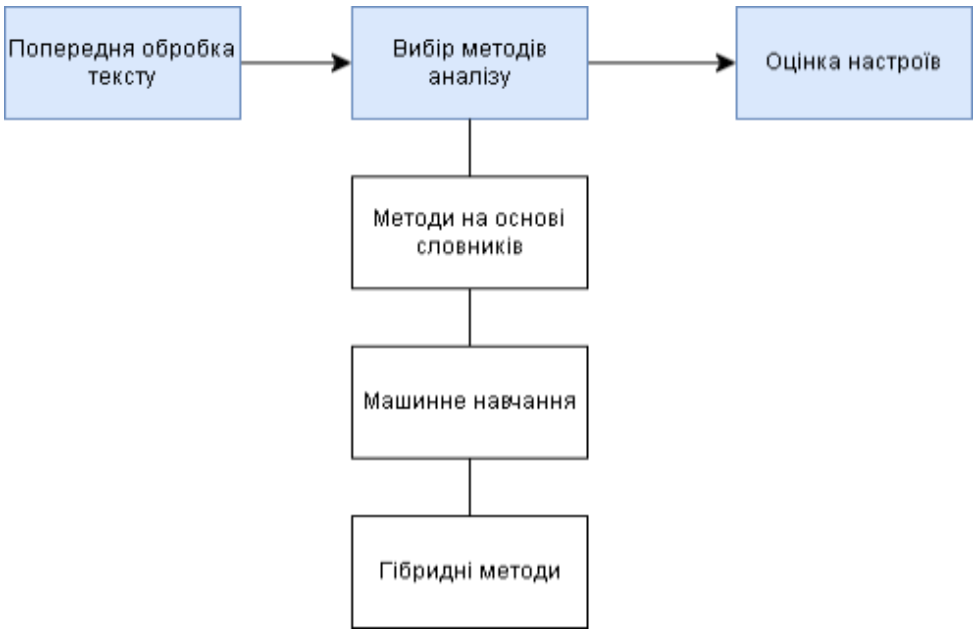

Рисунок 1.4 – Етапи аналізу настроїв

Попередня обробка тексту. Текст очищується від зайвих символів, таких як знаки пунктуації, цифри, спеціальні символи тощо. Це необхідно для зменшення шуму в даних.

Вибір методів аналізу. Є кілька методів для виконання аналізу настроїв. Це можуть бути:

– методи на основі словників: використовують набір попередньо визначених слів, що мають позитивне або негативне забарвлення. Наприклад,

система може оцінити наявність слів типу "щасливий", "погано", "злий" у тексті та зробити висновок про настрій;

– машинне навчання: використовуються алгоритми, що навчаються на великій кількості текстових даних для автоматичного визначення настроїв. Наприклад, за допомогою класифікації тексту можна автоматично навчити модель розрізняти позитивні та негативні емоції на основі прикладів;

– гібридні методи: поєднують словники та машинне навчання для досягнення більш точних результатів.

Оцінка настроїв. Результатом аналізу є класифікація тексту на одну з категорій:

– позитивний, якщо текст має позитивне емоційне забарвлення;

– негативний, якщо текст містить негативні емоції;

– нейтральний, якщо текст не виражає явних емоцій.

Для оцінки настроїв у реальному часі застосовуються алгоритми, що здійснюють пошук ключових емоційних індикаторів у текстах, таких як VADER (Valence Aware Dictionary and sEntiment Reasoner), який є одним із популярних інструментів для аналізу настроїв в англомовному контексті [8, 9].

Аналіз настроїв є важливою складовою системи для виявлення негативних тенденцій в соціальних мережах. З його допомогою можна виконувати дії:

– моніторити емоційний стан суспільства. Визначення загального настрою користувачів на певну подію чи проблему;

– розпізнавати дезінформацію. Оцінка емоційної маніпуляції в текстах, які можуть бути частиною кампанії з дезінформації.

У роботі [10] обговорюється важливість використання алгоритмів машинного навчання для ефективного виявлення мови ненависті.Аналіз настроїв допомагає зберігати баланс в онлайнових спільнотах і підтримує безпеку користувачів.

1.3 Огляд наявних аналогів та їхніх недоліків

У сучасному світі зростає кількість систем і платформ, що використовують автоматизовані методи аналізу текстового контенту з метою виявлення негативних тенденцій у соціальних мережах. Для вирішення таких проблем, як мова ненависті, кібербулінг і дезінформація, існує цілий ряд існуючих рішень, що використовують різноманітні підходи до обробки даних і аналізу настроїв. Проте, навіть наявні технології мають свої обмеження та недоліки, що потребують удосконалення.

Існують різноманітні інструменти, бібліотеки та платформи, що дозволяють здійснювати аналіз контенту на основі NLP та аналізу настроїв.

Hatebase. Платформа для виявлення мови ненависті в текстах. Hatebase використовує великий словник термінів, що відносяться до мови ненависті, і дозволяє автоматично класифікувати коментарі та пости на платформах, таких як Twitter чи Reddit. Однак цей інструмент є обмеженим у плані контекстного аналізу, оскільки він базується на використанні попередньо визначених термінів, що можуть не охоплювати всі варіації висловлювань.

Рисунок 1.5 – Платформа Hatebase

Google Perspective API. Це інструмент, розроблений компанією Jigsaw (підрозділом Google), для оцінки токсичності текстів. Perspective API оцінює рівень агресії чи токсичності коментаря на основі певних характеристик, таких як тон, критика чи тролінг. Хоча Perspective API використовує машинне навчання для оцінки токсичності, система має свої обмеження в точності, оскільки може неправильно трактувати контекст або невірно оцінювати сарказм чи іронію в коментарях.

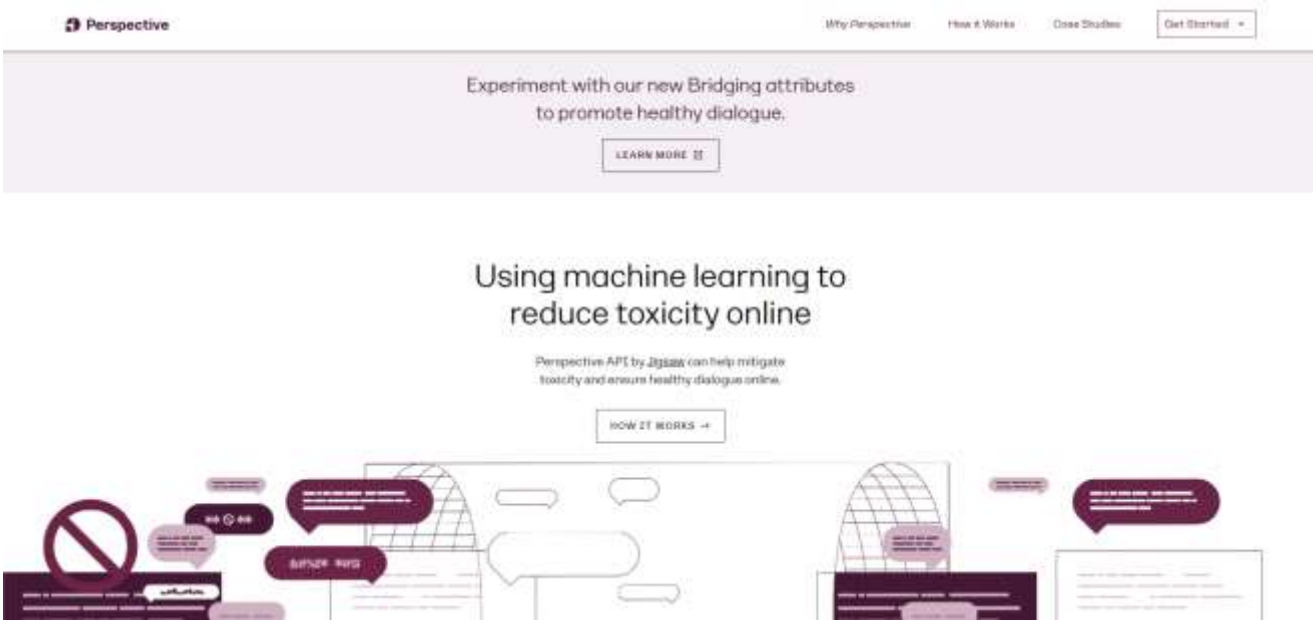

Рисунок 1.6 – Google Perspective API

IBM Watson Natural Language Understanding є потужною платформою для глибинного аналізу тексту, яка застосовує машинне навчання для визначення настроїв, емоцій і контексту текстових даних. Використовуючи розширені моделі, Watson здатний виявляти контекстуальні неточності та класифікувати текст за різними категоріями, що робить його ефективним інструментом для виявлення негативних тенденцій у соціальних мережах. Однак для ефективного функціонування Watson вимагає значних обчислювальних ресурсів та спеціалізованих налаштувань, особливо при обробці великих обсягів даних, що може створювати певні технічні виклики.

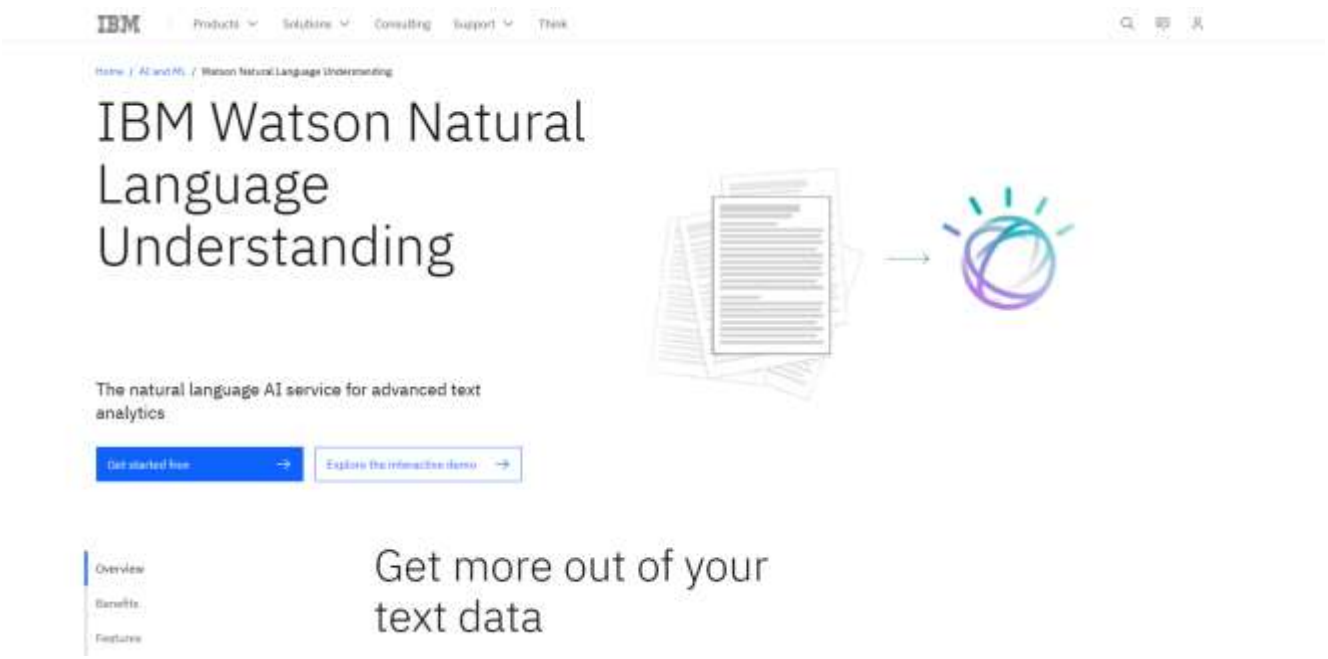

Рисунок 1.7 – IBM Watson Natural Language Understanding

Ці інструменти мають свої переваги, але також стикаються з обмеженнями, що знижують точність результатів при роботі з великими обсягами текстів або складними контекстами. Наприклад, попередньо визначені терміни можуть не охоплювати нові варіації мови, а також виникають труднощі з точним визначенням настрою в складних текстах, таких як сарказм чи іронія. У зв'язку з цим важливо розробляти системи, які здатні більш гнучко і точно аналізувати соціальні медіа та інші типи текстового контенту.

1.4 Проблеми і виклики у виявленні негативного контенту

Аналіз та виявлення негативного контенту в соціальних мережах є складним і багатоаспектним завданням. Основні труднощі пов'язані з особливостями мови користувачів, швидкими змінами контенту та необхідністю точних і ефективних інструментів для автоматизованого аналізу.

1.4.1 Неоднозначність контенту

Однією з основних проблем є неоднозначність текстового контенту, що ускладнює автоматизоване виявлення негативних елементів. Часто негативні

висловлювання можуть бути приховані у контексті жартів, сарказму або іронії. Стандартні алгоритми аналізу настроїв, що спираються на лексичний аналіз, мають труднощі з розпізнаванням таких складних контекстів. У статті [11] підкреслюється важливість контексту при виявленні сарказму в текстах соціальних мереж, що є складним завданням для стандартних алгоритмів аналізу настроїв. Наприклад, фраза з позитивним лексичним змістом може бути саркастичною, що вимагає врахування більш широкого контексту, ніж простий аналіз слів. "Це неймовірно чудово, коли забуваєш свій телефон вдома", що на перший погляд виглядає як позитивний відгук, але з урахуванням контексту розмови чи відтінку сарказму насправді може означати негативне ставлення до ситуації. Алгоритм, заснований лише на лексичному аналізі, може неправильно оцінити такий контент як нейтральний або навіть позитивний. Це потребує використання складніших моделей глибокого навчання, таких як BERT чи GPT, які здатні краще обробляти контекст і визначати відтінки значень, що дає змогу точніше виявляти негативний контент, навіть коли він виражений непрямо або через сарказм.

1.4.2 Динамічність мови та сленг

У дослідженні [12] розглядаються труднощі, що виникають через постійну зміну сленгових виразів та мемів у соціальних мережах, що ускладнює точний аналіз контенту. Користувачі соціальних мереж часто використовують неформальні мовні конструкції, включаючи скорочення, сленг, нові терміни, емодзі та меми. Ці елементи швидко змінюються, що ускладнює створення ефективних моделей для виявлення негативного контенту. Системи, засновані на фіксованих словниках або моделях, можуть не враховувати нові форми вираження або змінені значення слів. Наприклад, слово, яке раніше мало нейтральне значення, може отримати нове, образливе забарвлення у контексті певної спільноти. Мова користувачів соціальних мереж швидко змінюється, постійно з'являються нові сленгові слова, абревіатури, меми та символи, які важко постійно відслідковувати та додавати до алгоритмів аналізу. Крім того, вживання емодзі або специфічних

символів може суттєво змінювати тон коментаря. Наприклад, комбінація тексту і емодзі може надати образливий характер повідомленню. Для цього сучасні системи мають враховувати не тільки текст, але й візуальні елементи (наприклад, емодзі), що ускладнює процес класифікації. У сучасних мережах популярні вирази типу "це просто флекс", де слово "флекс" вказує на демонстрацію чогось позитивного, однак в іншому контексті цей термін може використовуватися з негативним підтекстом. Якщо система аналізує його за старим словником, вона може не розпізнати новий зміст і трактувати його як нейтральний. Таким чином, системи виявлення негативного контенту повинні постійно оновлюватися, щоб бути в курсі нових тенденцій і термінів, що використовуються в соціальних мережах.

1.4.3 Культурні та мовні особливості

Іншою важливою проблемою є культурні та мовні відмінності, які можуть впливати на інтерпретацію контенту. Те, що в одній культурі може вважатися образливим або неприязним, у іншій може сприйматися як звичайна частина комунікації. Це створює складнощі для універсальних моделей виявлення негативного контенту, оскільки такі моделі повинні враховувати специфіку мови та культури кожного користувача.

У дослідженні [13] наголошується на важливості врахування культурних та мовних відмінностей при автоматизованому аналізі контенту, що є необхідним для точного виявлення негативних висловлювань. Також різні мови мають свої граматичні і лексичні особливості, що ускладнює створення багатомовних моделей для аналізу. Наприклад, одна й та сама фраза може мати різні конотації в залежності від культурного контексту, що потребує точного врахування цих нюансів. Для більш точної ідентифікації негативного контенту в таких умовах потрібно використовувати моделі, які адаптовані до специфіки мови і культури кожного користувача, а також системи, що можуть виявляти зміни у відносинах до слів чи виразів з часом.

1.4.4 Фальшиві новини та дезінформація

Дезінформація, або фальшиві новини, є особливою формою негативного контенту, яка має на меті маніпулювання громадською думкою або створення соціальних конфліктів. У роботі [14] обговорюється складність виявлення дезінформації через її схожість із правдивими або напівправдивими фактами, що ускладнює застосування автоматизованих методів перевірки фактів. Оскільки фальшиві новини часто подаються як правдиві або напівправдиві факти, їх виявлення за допомогою автоматизованих систем є складним завданням. Фальшиві новини, які стверджують, що "вчені підтвердили, що куріння є корисним для здоров'я", на перший погляд можуть виглядати як правдиві або полемічні, але після перевірки джерел можна виявити, що інформація є дезінформацією. Виявлення дезінформації потребує більш складних моделей, які можуть аналізувати не тільки сам текст, але й джерела інформації, перевіряти його на відповідність до надійних фактів і використовувати контекстуальні знання для ідентифікації неточностей. Для цього необхідна інтеграція кількох рівнів аналізу: семантичного, перевірки джерел і порівняння з надійними інформаційними ресурсами. Це потребує складної інфраструктури, що може аналізувати велике число джерел і контекстуальних даних, а також значних ресурсів для обробки і перевірки кожного джерела.

1.4.5 Захист свободи слова

Один із ключових викликів при виявленні негативного контенту – це забезпечення балансу між виявленням мови ненависті та захистом свободи слова. При автоматизованій фільтрації контенту існує ризик, що система може неправильно класифікувати нейтральні або навіть позитивні висловлювання як негативні, що може призвести до цензури або обмеження права на вільне вираження думок. У статті [15] розглядається проблема балансування між виявленням мови ненависті та захистом свободи слова, що є важливим аспектом при автоматизованому фільтруванні контенту.Це особливо складно, коли йдеться

про тонкі відмінності між критикою і неприязню. "Можна було б розглянути це питання по-іншому, але важко з такою системою" — подібний коментар може бути класифікований як негативний, навіть якщо це не є ненавистю, а просто критикою державної політики чи суспільної ситуації. Тому важливо розробляти точні моделі, які не тільки виявляють негативний контент, але й враховують контекст, щоб уникнути фальшивих позитивних спрацьовувань і забезпечити користувачам свободу вираження думок.

1.4.6 Машинне навчання та обмеження даних

Для створення ефективних моделей виявлення негативного контенту потрібні великі обсяги даних для навчання. У дослідженні [16] розглядаються проблеми, пов'язані з обмеженістю даних для навчання моделей машинного навчання для виявлення негативного контенту, що може призводити до помилок у класифікації. Однак зібрати достатню кількість даних, що точно відображають різноманітні негативні тенденції, досить складно. Моделі машинного навчання потребують як позитивних, так і негативних прикладів для навчання, але знайти збалансовані набори даних не завжди можливо. Алгоритм, що навчається на великій кількості тролінгових коментарів (наприклад, "Ти що, зовсім ненормальний?"), може мати складнощі з точністю у випадках, коли потрібно розрізняти справжнє тролінгування від м’якої критики або жарту. Це проблема обмеження даних, коли необхідно навчити систему правильно розпізнавати такі тонкі різниці. Це може призводити до упередженості алгоритмів або недостатньої точності в реальних умовах. Виявлення негативного контенту в соціальних мережах Для підвищення ефективності таких систем необхідно використовувати складні моделі глибокого навчання, які можуть враховувати контекст, змінювану природу мови і соціальні фактори, а також постійно оновлювати моделі відповідно до нових тенденцій у мережах.

Висновки до розділу 1

У першому розділі було досліджено ключові аспекти, що стосуються аналізу негативних тенденцій у соціальних мережах, зокрема таких явищ, як мова ненависті, кібербулінг та дезінформація. Визначено основні негативні явища, що виникають в інтернет-просторі, та їхній вплив на психоемоційний стан користувачів, а також на суспільство в цілому. Досліджено важливість своєчасного виявлення та протидії цим проблемам для забезпечення безпечного інформаційного середовища.

У рамках розділу також було розглянуто різноманітні методи обробки текстового контенту, зокрема обробку природної мови (NLP) та аналіз настроїв, які є основою для виявлення негативних тенденцій. Визначено роль таких методів як VADER, Perspective API, а також можливості використання методів машинного навчання для підвищення точності та адаптивності системи до нових форм маніпуляцій, таких як сарказм або спотворене подання інформації.

Щодо аналізу наявних аналогів, було проведено огляд основних інструментів, що використовуються для виявлення негативного контенту у соціальних мережах. Серед основних проблем таких рішень виділено обмежену точність у розпізнаванні контексту та культурних відмінностей, високі вимоги до обчислювальних ресурсів, а також нездатність ефективно працювати з новими формами маніпуляцій.

2 ЕТИЧНІ ПИТАННЯ ТА ПРАВОВІ АСПЕКТИ ВИЯВЛЕННЯ НЕГАТИВНОГО КОНТЕНТУ

2.1 Етика виявлення негативного контенту

Виявлення негативного контенту в соціальних мережах автоматичними системами постає перед низкою етичних викликів. Перш за все, важливо розглянути питання визначення того, що є "негативним" або "неправильним" контентом, а також, чи можна автоматизувати цей процес без втручання людини, враховуючи культурні, мовні та соціальні особливості.

Автоматизація виявлення негативного контенту полягає в використанні алгоритмів машинного навчання, що аналізують текст, зображення, відео і навіть аудіо для виявлення неприязних або ворожих висловлювань [16] . Це дозволяє обробляти великі обсяги даних в реальному часі. Однак автоматичне виявлення негативного контенту має свої ризики:

- контекстуальні помилки: алгоритми можуть не враховувати контекст коментаря або публікації, що призводить до хибної класифікації. Наприклад, використання сарказму чи іронії може бути помилково визнано негативним контентом;
- складність з багатозначними словами: слова або фрази, які можуть мати різне значення в різних контекстах, потребують особливо обережного підходу для їх трактування.

Автоматизація виявлення негативного контенту без втручання людини не є остаточним рішенням. Важливо забезпечити баланс між швидкістю автоматизованих систем і точністю класифікації, що можливо лише через поєднання автоматичних алгоритмів з людським контролем.

Одним із важливих аспектів етики є забезпечення неупередженості системи виявлення негативного контенту. Алгоритми можуть бути схильні до упереджень, якщо вони були навчені на даних, що містять соціальні чи культурні стереотипи. Так, наприклад:

– упередження щодо певних груп: якщо тренувальний набір даних містить переважно тексти, написані людьми з певної культури або політичної орієнтації, система може проявляти упередженість до іншої групи;

– вибіркове трактування термінів: моделі, що використовують обмежену кількість даних, можуть неправильно інтерпретувати певні вирази або ідеї, які є культурно або мовно специфічними.

Таким чином, важливо забезпечити, щоб алгоритми були протестовані на багатьох культурних та мовних контекстах для мінімізації упереджень. Це дозволяє створити систему, яка об'єктивно визначає негативний контент, не виходячи за межі етичних норм.

Різні країни та культури можуть мати різні уявлення про те, що є негативним контентом. Те, що для однієї спільноти може бути прийнятним вираженням думок, для іншої може бути визнано неприязним або шкідливим. Важливо визначити критерії, за якими контент вважається негативним:

– ненависть та дискримінація: контент, який закликає до насильства, расизму, сексизму, чи будь-якої іншої форми дискримінації, є негативним і має бути видалений або оброблений;

– насильство : зображення або відео, що містять насильство або інші форми експлуатації, повинні бути негайно виявлені і заблоковані;

– дезінформація: фальшиві новини та маніпуляції, що можуть мати руйнівний вплив на суспільство, також повинні виявлятися системою.

Проте необхідно чітко визначити межі "негативного контенту", щоб не порушувати права на свободу вираження думок і не цензурувати конструктивні дискусії. Відсутність універсальних стандартів може призвести до небажаних наслідків, таких як надмірне цензурування або маніпуляції інформацією [17].

Виявлення і блокування негативного контенту може мати серйозні моральні наслідки для користувачів соціальних мереж. Якщо система помилково класифікує пост або коментар як негативний, це може призвести до обмеження свободи вираження, заборони на участь у спільноті або навіть до юридичних наслідків.

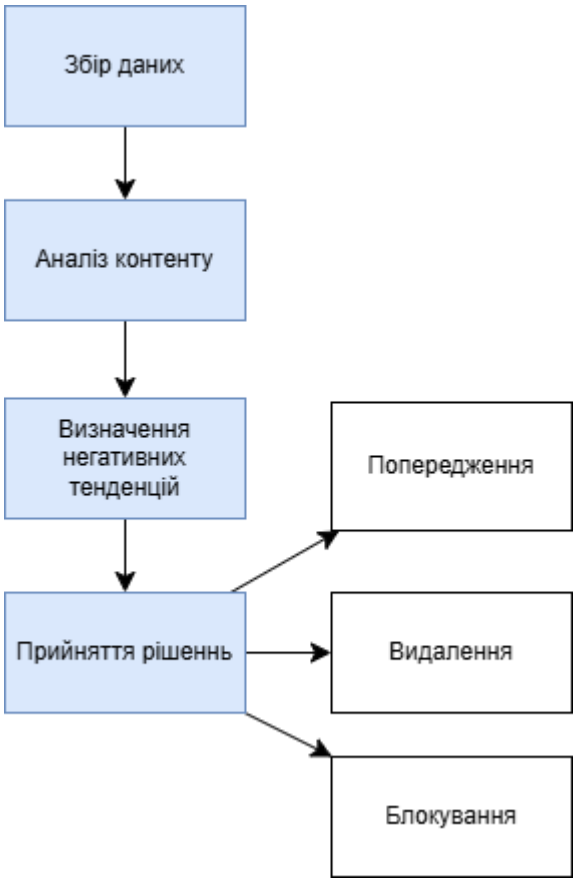

Рисунок 2.1 – Етапи виявлення негативного контенту

Крім того, блокування контенту може мати негативний вплив на репутацію користувача або організації, яка може бути позбавлена можливості висловити свою точку зору.

Отже, етика виявлення негативного контенту полягає в знаходженні правильного балансу між автоматизацією процесу і людським контролем, забезпеченням об'єктивності та запобіганням упередженості, а також чітким визначенням меж того, що вважається "негативним" контентом.

2.2 Правове регулювання шкідливого контенту

Виявлення та аналіз негативного контенту в соціальних мережах має значні правові наслідки, які потребують ретельної уваги. Ось кілька ключових правових аспектів, які слід враховувати під час автоматичного виявлення та обробки контенту.

Згідно з європейським законодавством, зокрема з Регламентом (ЄС) 2016/679 (GDPR), компанії, які збирають та обробляють персональні дані користувачів, повинні дотримуватися строгих вимог щодо захисту цих даних [18]. Під час автоматичного виявлення негативного контенту, особливо якщо збираються та обробляються дані про осіб (наприклад, коментарі, пости, повідомлення), необхідно дотримуватись наступних вимог:

– прозорість: користувачі повинні бути поінформовані про збір їхніх персональних даних та цілі обробки;

– згода: обробка персональних даних має бути здійснена за згодою користувачів, особливо у випадках, коли дані використовуються для специфічних цілей, таких як аналітика або створення профілів;

– мінімізація даних: збір даних має бути обмежений лише тим, що необхідно для досягнення цілей обробки;

– безпека даних: має бути вжито заходів для захисту даних від несанкціонованого доступу та зловживань.

Різні країни почали приймати законодавчі ініціативи, що вимагають від соціальних мереж впровадження автоматичних систем для виявлення і видалення незаконного або шкідливого контенту.

Німеччина прийняла закон про мережеву прозорість (NetzDG), який зобов'язує платформи видаляти або блокувати шкідливий контент, наприклад, мову ворожнечі та тероризм, у терміни, що не перевищують 24 години.

Європейський Союз: розробляються нові правила для соціальних платформ (наприклад, Digital Services Act), які накладають на компанії обов'язки щодо прозорості алгоритмів та підзвітності за контент, що публікується.

Правові аспекти виявлення негативного контенту є важливими для забезпечення справедливості, законності та захисту прав користувачів.

2.3 Проблеми з приватністю

Однією з основних проблем є збір, обробка та зберігання особистої інформації користувачів, яка може включати не тільки текстові повідомлення, а й інформацію про місцезнаходження, інтереси, активність у мережі тощо. Якщо система не дотримується належних стандартів безпеки та приватності, це може призвести до витоку даних.

Багато соціальних мереж мають складні та незрозумілі для середнього користувача політики конфіденційності. Користувачі часто не усвідомлюють, яка саме інформація може бути використана в процесі збору даних. В результаті виникає ризик порушення прав на приватність.

Хоча анонімність в інтернеті може бути важливою для захисту приватності, вона також може бути використана для поширення шкідливого контенту. Проблеми з анонімністю можуть призвести до відсутності відповідальності за негативний контент, такий як кібербулінг чи мова ворожнечі [19].

В деяких випадках соціальні мережі збирають дані користувачів без їхньої явної згоди, що викликає серйозні занепокоєння щодо порушення прав на приватність. Це включає автоматичний збір даних про поведінку користувачів через трекери або "cookies".

Проблеми приватності ускладнюються, коли соціальні мережі працюють на глобальному рівні, де дія різних законів про захист даних (наприклад, GDPR в Європейському Союзі, CCPA в Каліфорнії) може суперечити одна одній. Тому системи, що аналізують контент, мають бути адаптовані до міжнародних стандартів [20].

Висновки до розділу 2

У розділі 2 було розглянуто ключові етичні та правові аспекти, що виникають у процесі виявлення негативного контенту в соціальних мережах. Автоматизація цього процесу, хоч і дозволяє обробляти великі обсяги даних, стикається з низкою викликів. По-перше, це етичні проблеми, пов’язані з контекстуальними помилками, упередженням систем та відсутністю універсальних стандартів. Ефективне виявлення контенту потребує поєднання алгоритмів машинного навчання з людським втручанням для досягнення об'єктивності та мінімізації помилок.

Правові аспекти зосереджені на забезпеченні захисту персональних даних, дотриманні норм конфіденційності та відповідності міжнародним стандартам, таким як GDPR. Проблеми приватності, пов’язані з анонімністю, політиками конфіденційності та можливим збором даних без згоди користувачів, залишаються основними викликами для соціальних мереж.

Отже, для ефективного і етичного виявлення негативного контенту необхідно розробляти комплексні стратегії, що враховують як етичні, так і правові аспекти. Це дозволить не тільки забезпечити безпеку користувачів, але й зберегти їхні права на приватність та свободу вираження.

3 МЕТОДИ ТА МОДЕЛІ ДЛЯ АНАЛІЗУ КОНТЕНТУ

3.1 Опис та обґрунтування вибору методів аналізу настроїв

Аналіз настроїв є одним з основних напрямів досліджень в області обробки природної мови (NLP), що має на меті визначити емоційне забарвлення тексту: позитивне, негативне або нейтральне. Для вирішення задачі виявлення негативних тенденцій у соціальних мережах, таких як мова ненависті, кібербулінг чи дезінформація, необхідно застосовувати надійні й ефективні методи аналізу настроїв. У цьому підрозділі розглянуті основні методи та моделі, які обрано для реалізації в рамках даного дослідження, а також обґрунтування їх вибору.

Аналіз настроїв зазвичай ґрунтується на двох основних підходах: правиловому (лексико-семантичному) та машинному навчанні.

Правиловий підхід використовує лексичні ресурси, такі як словники позитивних і негативних слів (наприклад, VADER, SentiWordNet) для визначення емоційної оцінки тексту. Це підхід зручний для аналізу невеликого обсягу тексту або коли важливі деталі лексичного значення.

Аналіз настроїв вимагає методів, які можуть працювати з різними типами текстових даних (наприклад, короткими повідомленнями, коментарями чи постами на форумах), що мають специфічні особливості, як-от:

- короткі тексти. Тексти в соціальних мережах часто є короткими (наприклад, коментарі в 140-280 символів, твіти), що може бути складним для традиційних методів аналізу тексту, які не завжди можуть точно вловити емоційний зміст таких повідомлень;
- контекстуальність. В соціальних мережах широко використовуються сарказм, іронія, а також сленг, що ускладнює процес аналізу настроїв. Багато стандартних методів можуть не виявляти таких нюансів;
- різноманітність тем. Коментарі можуть стосуватися різноманітних тем (від політики до повсякденних подій), і для цього важливо використовувати методи, які здатні адаптуватися до різних контекстів.

Для кваліфікаційної магістерської роботи вибір методів аналізу настроїв обґрунтовується необхідністю забезпечення високої точності при роботі з соціальними мережами. Підходи, які найбільше відповідають вимогам сучасних технологій аналізу великих даних та дозволяють враховувати складні контексти, є найкращими для цілей цього дослідження.

Для дослідження настроїв у соціальних мережах було обрано три методи: VADER, BERT та TextBlob. Кожен із цих методів має свої особливості та застосування, які дозволяють отримати максимально точні результати в залежності від типу тексту та його контексту.

Одним із найбільш популярних методів для аналізу настроїв є лексиконний підхід. VADER є популярним інструментом для лексиконного аналізу настроїв, розробленим спеціально для роботи з текстовими даними з соціальних мереж. Метод VADER (Valence Aware Dictionary and sEntiment Reasoner) є словниково-орієнтованим аналізатором настроїв. У статті [21] розглянуто переваги використання VADER для аналізу настроїв у текстах соціальних мереж, зокрема у коротких коментарях. Цей метод оцінює текст за допомогою попередньо визначених словників, що містять позитивні, негативні та нейтральні слова, а також правила для обробки різних контекстів, таких як емоційні інтонації або смайлики.

VADER часто використовується для аналізу настроїв у коротких повідомленнях соціальних мереж, таких як твіти або коментарі. Наприклад, фраза "I LOVE this movie!!!" буде класифікована як позитивна з високою емоційною інтенсивністю завдяки використанню слова "LOVE", великим літерам і знакам оклику. Коментар з високою емоційною інтенсивністю в інформаційній системі наведений нижче.

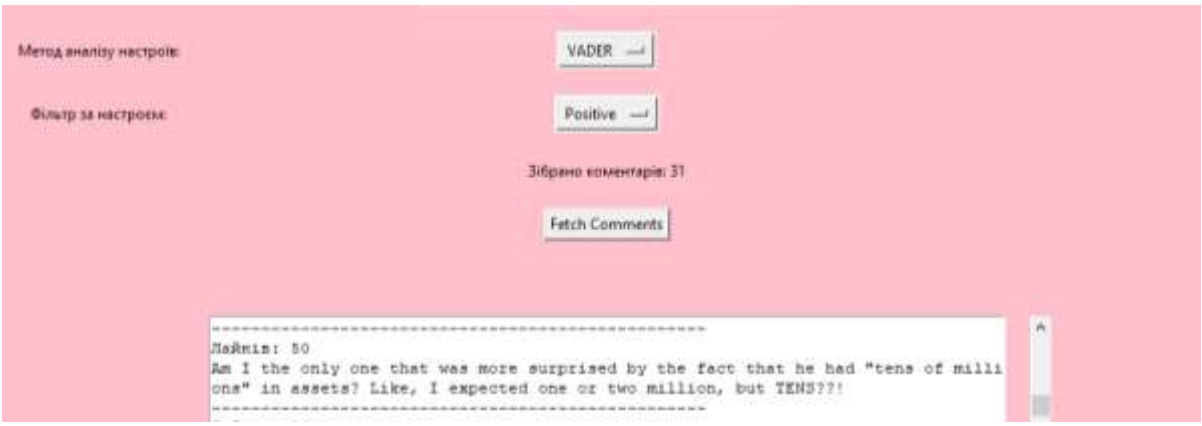

Рисунок 3.1 – Коментар з високою емоційною інтенсивністю

Рисунок 3.2 – Переваги та недоліки методу VADER

Для більш точного аналізу, особливо при роботі з великими та складними наборами даних, застосовуються методи глибокого навчання, такі як BERT (Bidirectional Encoder Representations from Transformers). BERT використовує трансформерні мережі, здатні навчатися з контексту обох напрямків (ліворуч і праворуч), що дозволяє краще інтерпретувати складні тексти з емоційними

відтінками. BERT є потужною моделлю глибокого навчання, що дозволяє аналізувати текст у двох напрямках одночасно – зліва направо і справа наліво. Це дає змогу моделі краще розуміти контекст, враховуючи попередні та наступні слова у реченні. Завдяки своїй здатності до глибокого навчання, BERT став одним із найбільш ефективних інструментів для аналізу настроїв та інших завдань обробки природної мови (NLP).

BERT використовує архітектуру трансформерів, яка відрізняється від традиційних рекурентних нейронних мереж (RNN) своєю здатністю обробляти тексти паралельно, що робить його більш ефективним для великих обсягів даних.

Рисунок 3.3 – Переваги та недоліки методу BERT

Як зазначено в [22], аналіз настроїв дозволяє виявити емоційне забарвлення повідомлень у соціальних мережах. Модель BERT може бути використана для аналізу настроїв у більш складних текстах, наприклад, в новинних статтях або довгих дискусіях в блогах. Вона здатна правильно розпізнати саркастичні висловлювання на кшталт "Ну звісно, це була найкраща ідея", де традиційні лексиконні методи можуть помилково класифікувати текст як позитивний.

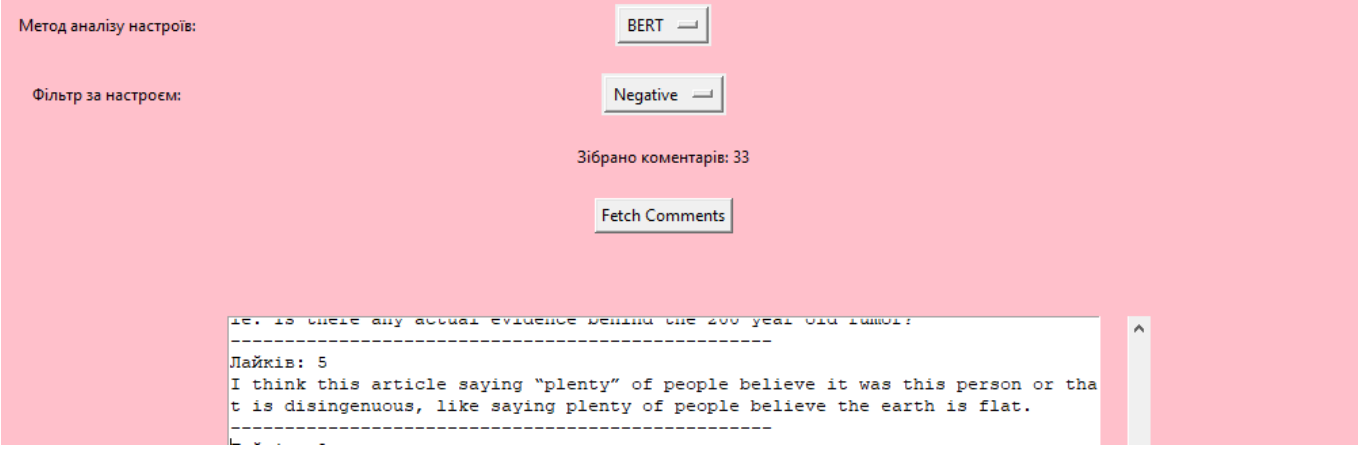

Рисунок 3.4 – Коментар , проаналізований моделлю BERT

Для вибору найбільш відповідного методу аналізу настроїв важливо розуміти, коли краще використовувати VADER, а коли BERT. Ось кілька ключових відмінностей між ними:

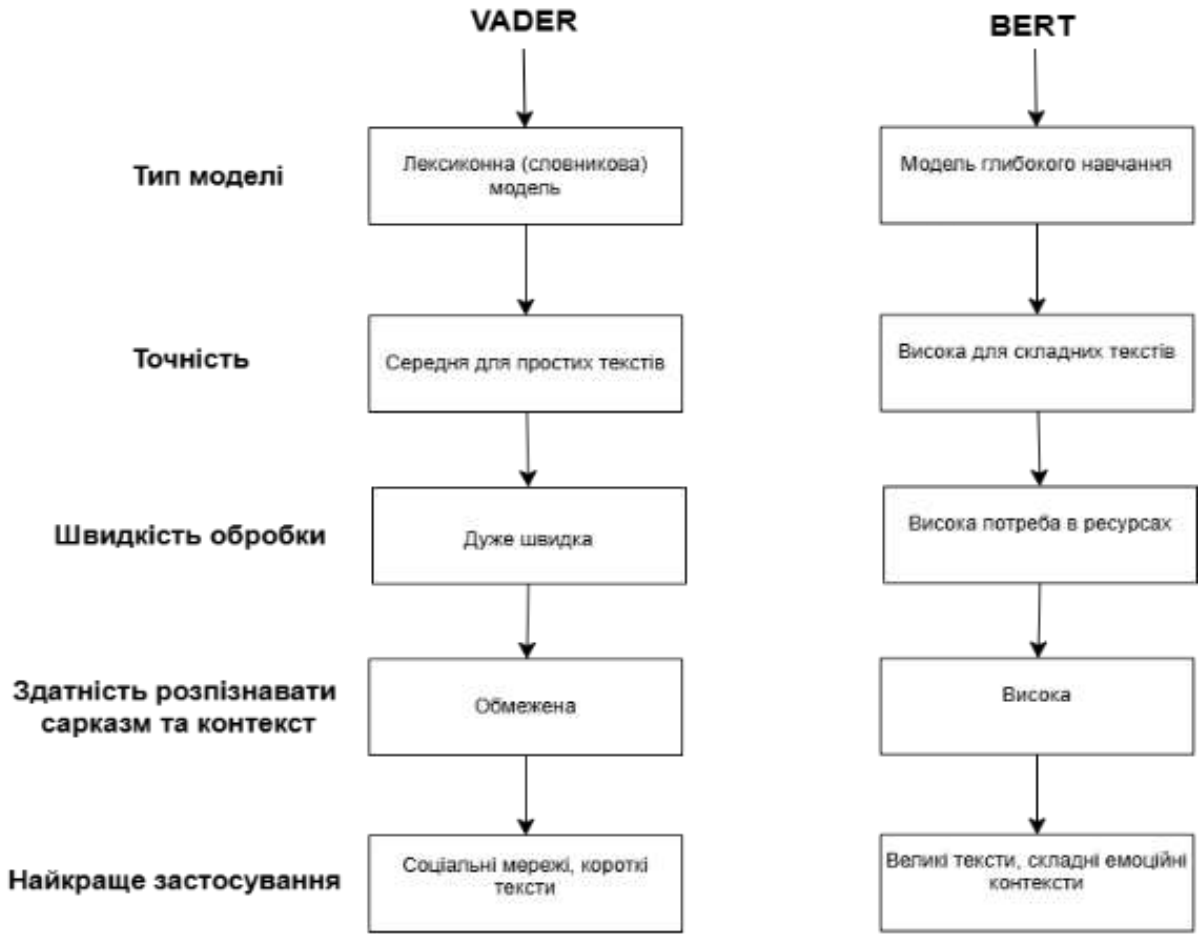

Рисунок 3.5 – Різниця між VADER і BERT

TextBlob є інструментом для обробки природної мови, який надає можливість визначати полярність (позитивна, негативна або нейтральна) та суб'єктивність тексту. Це інструмент підходить для обробки великих обсягів тексту з базовими вимогами до точності. Це набір інструментів для аналізу тексту, яка допомагає виконувати такі завдання, як токенізація, визначення частин мови, аналіз настроїв, іменовані сутності та інше. TextBlob побудована на базі популярних бібліотек NLTK і Pattern [23].

Основні можливості TextBlob включають не лише аналіз настроїв, але й визначення граматичних частин мови, переклад текстів та виправлення граматичних помилок. TextBlob також здатен здійснювати такі операції, як розбиття тексту на речення та слова, що робить його універсальним інструментом для обробки природної мови.

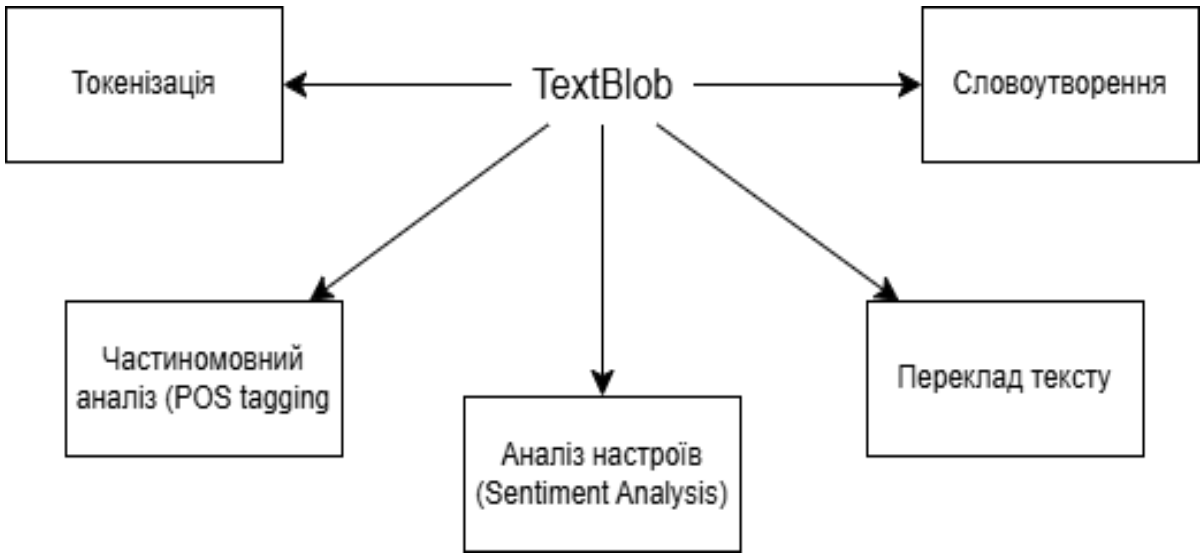

Рисунок 3.6 – Основні можливості TextBlob

Частиномовний аналіз (POS tagging) визначає частини мови для кожного слова у тексті. Аналіз настроїв (Sentiment Analysis) визначає полярність (негативний, нейтральний або позитивний тон) і суб'єктивність тексту (рівень об'єктивності чи суб'єктивності). Переклад тексту: можливість перекладати тексти

за допомогою API Google Translate. Словоутворення: перетворення слова у множину або однину, змінює час дієслів тощо.

Однією з ключових функцій є аналіз настроїв, яка оцінює тон тексту за допомогою лексиконного підходу. TextBlob використовує вбудовані словники з емоційними оцінками слів для оцінки полярності (від -1 до 1) і суб'єктивності (від 0 до 1), ось наприклад:

- полярність: від -1 (негативна) до 1 (позитивна);
- суб'єктивність: від 0 (об'єктивна) до 1 (суб'єктивна).

Рисунок 3.7 – Коментарі , проаналізовані моделлю TextBlob

TextBlob добре підходить для простих і швидких завдань, особливо в ситуаціях, коли не потрібна висока точність. TextBlob також може бути корисним для аналізу настроїв у невеликих текстових вибірках або при обробці даних із обмеженим обсягом, коли важливо швидко отримати результати [24]. Його простота та легкість у використанні роблять його популярним інструментом для прототипування та навчальних проєктів, де немає потреби в застосуванні складних алгоритмів або великих обчислювальних ресурсів.

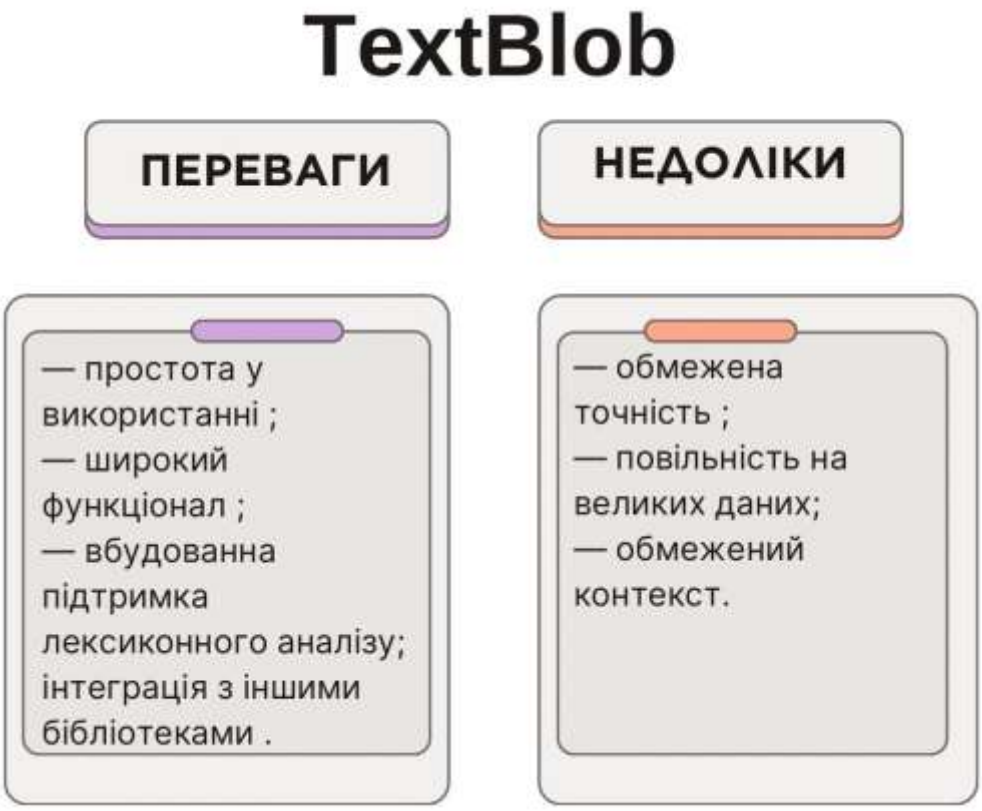

Рисунок 3.8 – Переваги та недоліки методу TextBlob

Перевагами TextBlob є:

– простота у використанні. TextBlob дуже легка в освоєнні, що робить її ідеальною для новачків у NLP. Вона не потребує складних налаштувань і легко інтегрується в проекти Python;

– легкість для невеликих проектів : TextBlob добре підходить для невеликих задач NLP, таких як аналіз коментарів або коротких текстів;

– широкий функціонал: бібліотека дозволяє виконувати багато різних завдань NLP: від аналізу настроїв до перекладу текстів;

– вбудована підтримка лексиконного аналізу: TextBlob використовує лексиконний підхід до аналізу настроїв, що дозволяє швидко аналізувати тексти без складних моделей машинного навчання;

– інтеграція з іншими бібліотеками: вона може використовувати ресурси з інших бібліотек, таких як NLTK та Pattern.

Недоліками TextBlob є:

– обмежена точність: оскільки TextBlob використовує лексиконний підхід, вона може бути менш точною в порівнянні з сучасними методами машинного навчання, такими як BERT або VADER. Вона може не розпізнавати сарказм, складні емоційні відтінки або контекст;

– повільність на великих даних: TextBlob підходить для аналізу невеликих текстів, але на великих обсягах даних може працювати повільно;

– обмежений контекст: аналіз настроїв TextBlob не враховує складний контекст речення або довгі залежності між словами, оскільки це лексиконний підхід. Це робить його менш точним для довгих текстів або більш складних конструкцій;

– застарілі алгоритми: на сьогодні існують більш сучасні підходи для обробки текстів, такі як трансформери (BERT), які краще справляються з аналізом настроїв у складних текстах.

Отже, TextBlob – це потужний інструмент для початкового рівня роботи з NLP, особливо для простих завдань, таких як аналіз настроїв або базова обробка тексту [25, 26].

3.2 Алгоритми роботи з текстовими даними

Обробка текстових даних за допомогою алгоритмів NLP (Natural Language Processing, обробка природної мови) є важливою частиною кваліфікаційної магістерської роботи, оскільки вона дозволяє аналізувати великі масиви текстової інформації з соціальних мереж, новинних ресурсів, блогів тощо. У межах аналізу настроїв ключовими завданнями є попередня обробка текстів, класифікація настроїв і використання відповідних алгоритмів для точного виявлення емоційного контексту [27] .

Перед тим як застосовувати алгоритми класифікації настроїв, необхідно провести попередню обробку текстових даних, яка включає такі етапи:

– токенізація: це процес розбиття тексту на окремі елементи (токени), зазвичай слова або речення. Це перший крок у більшості NLP-процесів, що дозволяє алгоритмам краще розуміти структуру тексту;

– нормалізація тексту: цей етап включає перетворення всіх символів у нижній регістр, видалення стоп-слів (найбільш частих слів, які не несуть значного смислового навантаження, наприклад "і", "або", "він"), а також видалення пунктуації та спеціальних символів;

– лематизація та стемінг: ці дві техніки використовуються для приведення слів до їхньої базової форми. Стемінг обрізає суфікси слів, залишаючи корінь (наприклад, "running" → "run"), тоді як лематизація повертає слово до його словникової форми на основі граматичного контексту (наприклад, "better" → "good").

Стемінг зменшує слова до кореня, що іноді може призвести до незрозумілих або некоректних форм. У результаті стемінгу слова "running" і "flying" перетворюються на "run" і "fli" відповідно, що може бути не зовсім граматично правильним.

Лематизація відновлює правильну граматичну форму слова, орієнтуючись на його значення в контексті. Наприклад, "better" лематизується в "good", а "studies" в "study".

Таблиця 3.1 – Порівняння стемінгу та лематизації

Слово оригінал	Стемінг	Лематизація
studies	studi	study
better	better	good
flying	fli	fly
running	run	run

Класифікація настроїв полягає у визначенні, чи є певний текст позитивним, негативним або нейтральним.

Рисунок 3.9 – Моделі класифікації настроїв

Для цього використовуються різні алгоритми, серед яких найбільш поширеними є:

- логістична регресія: це базовий підхід до класифікації настроїв, який оцінює ймовірність того, що даний текст належить до однієї з категорій настрою (позитивний, негативний, нейтральний) на основі певних характеристик тексту (наприклад, частоти вживання певних слів);
- модель на основі мішка слів (Bag of Words): ця модель перетворює текст у вектор числових значень на основі кількості появ кожного слова в тексті. Вона є простою, але ефективною технікою для представлення тексту в структурованій формі;

– TF-IDF (Term Frequency-Inverse Document Frequency): це вдосконалена модель, яка враховує як частоту слів у певному тексті (Term Frequency), так і рідкість вживання слова в усьому корпусі текстів (Inverse Document Frequency). Цей підхід дозволяє виділити значимі слова для кожного документа.

Для аналізу настроїв використовуються різні алгоритми машинного навчання та глибокого навчання [28]. Найпопулярніші з них описані нижче.

Наївний Байєс (Naive Bayes): це простий і швидкий алгоритм, який базується на ймовірностях. Він припускає незалежність між усіма характеристиками (словами) в тексті, що дозволяє швидко класифікувати текст на основі частотності слів.

Support Vector Machine (SVM): цей алгоритм використовується для класифікації текстів на основі визначення меж між різними класами (наприклад, позитивний чи негативний настрій). SVM будує гіперплощину, яка розділяє дані на два або більше класи.

Глибокі нейронні мережі, зокрема рекурентні нейронні мережі (RNN) та моделі на основі трансформерів, як BERT, стали важливими інструментами в аналізі тексту завдяки своїй здатності обробляти послідовності слів та враховувати складні контекстуальні зв'язки між ними. Такі моделі дозволяють точно класифікувати настрої, оскільки вони аналізують не тільки окремі слова, але й їх взаємодію в контексті. Наприклад, BERT використовує двосторонній контекст, що дає змогу краще розуміти значення слів залежно від їхнього оточення, що робить ці моделі особливо ефективними для виявлення емоцій, сарказму чи інших тонких нюансів в текстах.

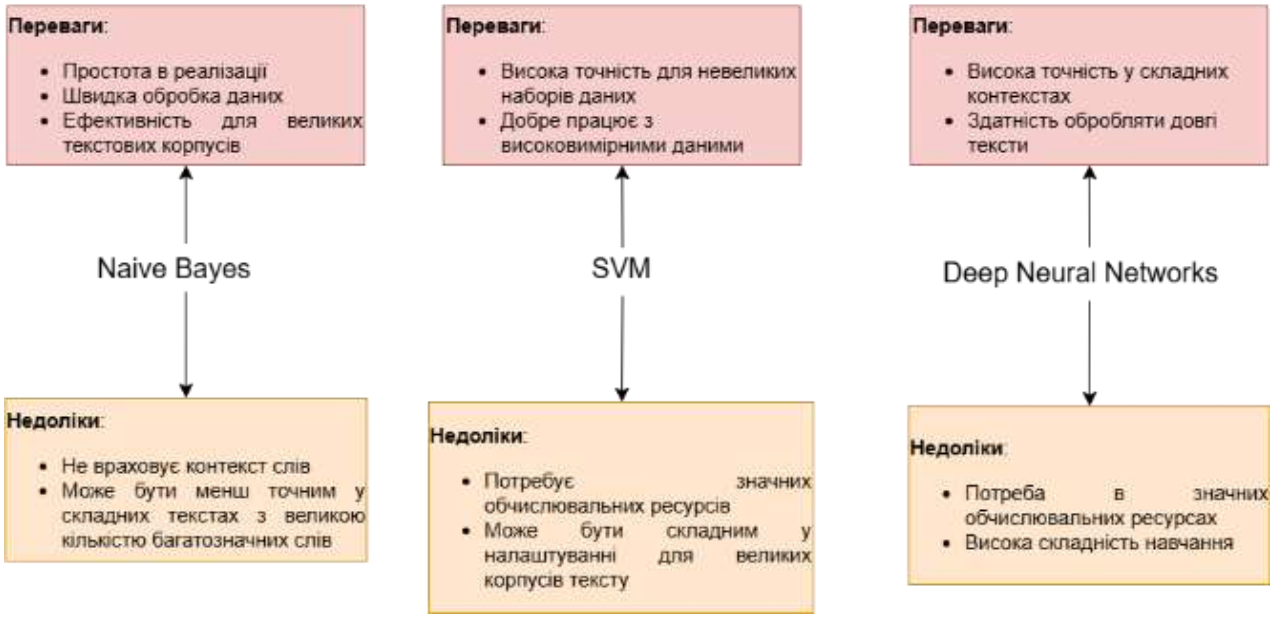

Рисунок 3.10 – Переваги та недоліки алгоритмів класифікації настроїв

Одним з найважливіших напрямків застосування NLP є виявлення мови ворожнечі та кібербулінгу, які є серйозними проблемами для соціальних мереж. У дослідженні [29] надано поради щодо передобробки текстових даних для подальшого використання в NLP-завданнях.

Методи класифікації. Для виявлення мови ворожнечі часто використовують класифікаційні алгоритми машинного навчання (Наївний Байес, SVM, логістична регресія), що працюють на основі розмічених даних. Кожен текст позначається як "мова ворожнечі" або "звичайний текст", після чого модель навчається розпізнавати ці категорії .

Глибинні нейронні мережі. Для складних задач, таких як виявлення прихованих або завуальованих форм мови ворожнечі (сарказм, іронія), застосовуються моделі глибинного навчання, зокрема BERT або GPT, які здатні розуміти складні контекстуальні взаємозв'язки між словами. У статті [30] обговорюється використання графіків для візуалізації результатів аналізу настроїв у реальному часі

Аналіз соціальних зв'язків. Крім текстового аналізу, важливим аспектом виявлення кібербулінгу є аналіз соціальних зв'язків користувачів у мережах.

3.3 Огляд бібліотек для аналізу та збору текстових даних

У даному підрозділі розглянуто три основні інструменти, які використовуються для розробки системи аналізу настроїв та збору текстових даних. Кожен із інструментів має свої специфічні особливості та підходить для різних задач в процесі розробки.

3.3.1 VADER

VADER є потужним та швидким інструментом для аналізу настроїв, який спеціалізується на обробці коротких текстів, таких як коментарі у соціальних мережах або твіти [18]. Його словниковий підхід дозволяє легко інтегрувати VADER у проекти, що займаються аналізом емоційного забарвлення текстів, і вимагають швидкого реагування на невеликі обсяги даних. Його основна перевага – здатність ефективно працювати з текстами, де відсутній глибокий контекст або складна граматика, що є характерним для коментарів у соцмережах.

У статті [31,32] йдеться про методи аналізу настроїв текстів у соціальних мережах, де розглядаються переваги VADER у порівнянні з іншими підходами до обробки коротких текстів. Ключові можливості VADER:

- швидка оцінка настроїв: VADER здатен оцінити текст за кількома основними категоріями: позитивний, негативний, нейтральний, а також визначити комбінований "compound score", який відображає загальний тон тексту;
- врахування контексту: VADER враховує такі фактори, як смайлики, пунктуацію (напр., знаки оклику) та підсилювачі (слова на кшталт "дуже", "абсолютно");
- орієнтованість на короткі тексти: VADER особливо ефективний для аналізу коротких текстів, де відсутні складні речення чи глибокий контекст, що є типовим для коментарів у соцмережах.

3.3.2 PRAW

PRAW (Python Reddit API Wrapper) є потужним інструментом для інтеграції Python з Reddit API. Ця бібліотека дозволяє зручно збирати дані з платформи Reddit, що є важливим джерелом для аналізу великих обсягів коментарів, постів і реакцій користувачів на різноманітні події або теми. Reddit є однією з найбільших платформ для обміну думками, що робить його особливо корисним для досліджень у сфері аналізу соціальних мереж. Переваги використання PRAW:

– легкість інтеграції: PRAW має високу гнучкість і підтримує всі основні функції API Reddit, що дозволяє ефективно взаємодіяти з даними, які генеруються на платформі;

– збір даних у реальному часі: за допомогою PRAW можна отримувати дані з популярних субредітів, що дозволяє аналізувати найактуальніші дискусії та події в режимі реального часу. Це особливо корисно для вивчення реакцій користувачів на новини або поточні події;

– гнучкість налаштувань: PRAW дозволяє здійснювати високий рівень налаштування для вибору певних категорій постів (наприклад, найпопулярніших чи найбільш обговорюваних), що дає змогу зібрати відповідний контент для подальшого аналізу.

Ця бібліотека дозволяє не лише збирати тексти, але й фільтрувати їх за різними параметрами, такими як кількість лайків, коментарів чи час публікації, що робить її зручним інструментом для обробки даних з Reddit [33].

3.3.3 Tkinter

Tkinter – це стандартна бібліотека Python для розробки простих графічних інтерфейсів користувача. Вона дозволяє створювати інтерфейси для взаємодії з користувачем, наприклад, для збору текстових даних або виведення результатів аналізу. Tkinter підтримує основні елементи інтерфейсу, такі як кнопки, текстові поля та меню, що дозволяє створювати базові програми без значних зусиль.

Рисунок 3.11 – Переваги та недоліки бібліотеки Tkinter

Усі три інструменти – VADER, PRAW та Tkinter мають свої специфічні застосування, які сприяють ефективному збору та аналізу даних. VADER є найкращим вибором для аналізу коротких текстів і оцінки емоційного забарвлення в реальному часі. PRAW дозволяє зручно інтегрувати Python з Reddit для збору великих обсягів даних і взаємодії з платформою, що є ключовим для соціальних мереж. Tkinter, у свою чергу, забезпечує простий і зручний спосіб створення інтерфейсів для користувачів, що дозволяє зручно взаємодіяти з даними та виводити результати аналізу [34].

Висновки до розділу 3

У другому розділі досліджено основні методи та алгоритми, які використовуються для аналізу настроїв тексту в соціальних мережах, зокрема для виявлення негативних тенденцій, таких як мова ненависті, кібербулінг та дезінформація. Описано два основні підходи до аналізу настроїв: правиловий та машинного навчання. Кожен з цих підходів має свої переваги та обмеження, але для вирішення задач, пов'язаних з великими обсягами тексту з соціальних мереж, найефективнішими є методи машинного навчання, зокрема глибоке навчання на основі трансформерних мереж.

Було обґрунтовано вибір методів для реалізації аналізу настроїв. Вказано на ефективність використання лексиконного підходу, такого як VADER, для аналізу коротких текстів, що широко застосовуються в соціальних мережах. Однак для більш складних текстів, таких як новини або довгі дискусії, рекомендується застосовувати методи глибокого навчання, зокрема BERT, який здатний враховувати контекст і емоційні відтінки тексту, що особливо важливо для правильної інтерпретації сарказму та іронії.

Також було розглянуто алгоритми обробки текстових даних та класифікації настроїв. Описано основні етапи попередньої обробки тексту, такі як токенізація, нормалізація, стемінг та лематизація. Ці етапи є важливими для створення якісних даних для подальшого аналізу. Розглянуті методи класифікації, зокрема логістична регресія, модель "мішок слів", TF-IDF, а також алгоритми машинного навчання, такі як наївний Байєс і Support Vector Machines (SVM), що дозволяють ефективно класифікувати тексти за настроєм.

4 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

4.1 Вибір соціальної мережі для реалізації інформаційної системи

Вибір соціальної мережі для реалізації інформаційної системи для аналізу контенту є критичним етапом, оскільки він визначає, яку платформу використовувати для збору, обробки та аналізу даних. У рамках даного дослідження було обрано **Reddit** як основну соціальну мережу для збору та аналізу контенту, оскільки вона найкраще відповідає вимогам проекту з урахуванням етичних, правових та технічних аспектів.

Рисунок 4.1 – Соціальна мережа Reddit

4.1.1 Відкритість та доступність даних

Однією з основних причин вибору Reddit є його відносна відкритість у наданні публічного контенту через API (Application Programming Interface). Reddit дозволяє отримувати доступ до постів, коментарів, а також різноманітної статистики (наприклад, кількість лайків, коментарів і репостів), що робить його ідеальним для збору текстових даних.

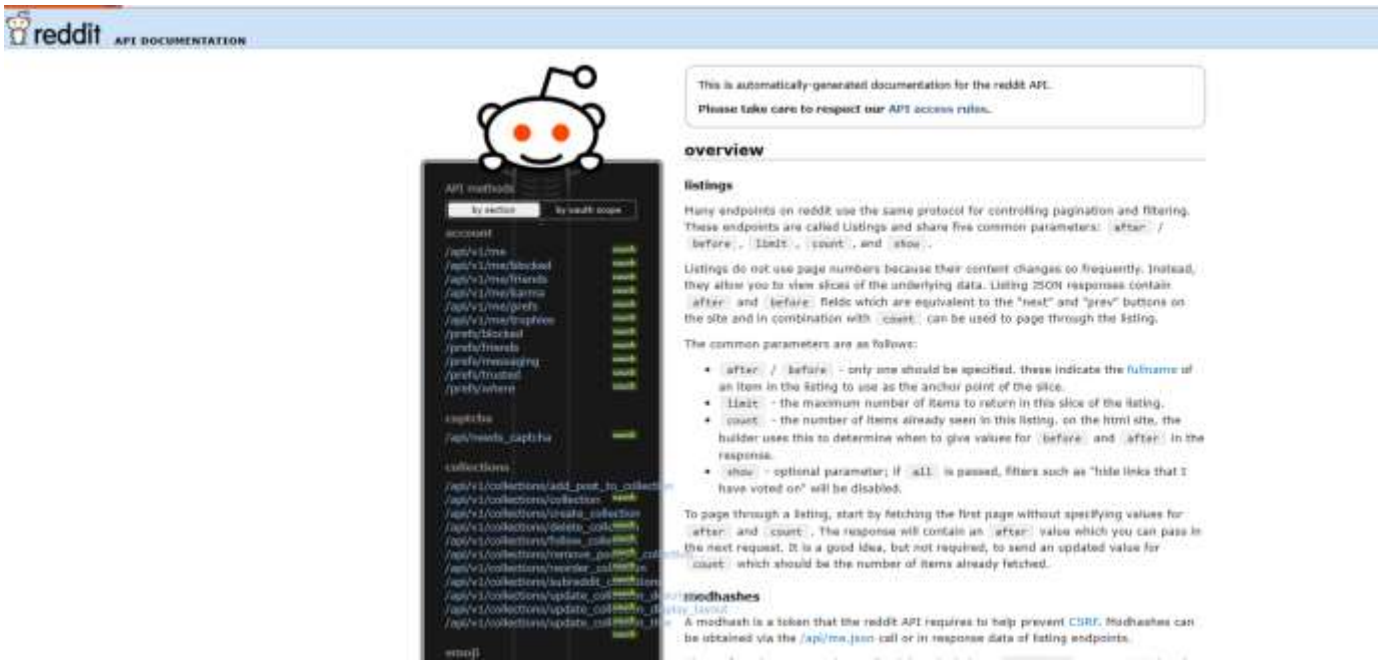

Рисунок 4.2 – Офіційний сайт для API Reddit

У документації можна знайти всі доступні методи для отримання постів, коментарів, користувацьких профілів тощо. Відкрите API Reddit дозволяє без значних обмежень збирати інформацію без порушення політики конфіденційності, що є важливим для забезпечення етичності дослідження.

4.1.2 Анонімність користувачів

Reddit є анонімною платформою, що дозволяє користувачам створювати акаунти під псевдонімами, без необхідності надавати особисті дані, що забезпечує високий рівень конфіденційності. Такий підхід значно знижує ризики порушення приватності порівняно з платформами, які вимагають реєстрацію через особисті акаунти, як-от Facebook або Instagram, де збирається більше персональних даних. Анонімність також дозволяє користувачам відкрито висловлювати свої думки без побоювань щодо персональної ідентифікації, що робить Reddit цінним джерелом для збору автентичних даних для аналізу соціальних тенденцій. Водночас, через відсутність ідентифікації, на платформі можуть виникати випадки поширення ненависницького контенту, що підкреслює важливість використання автоматизованих систем для виявлення та моніторингу таких негативних явищ.

4.1.3 Відсутність великої кількості чутливих даних

На відміну від таких соціальних мереж, як Facebook та Instagram, що активно збирають персональні дані для таргетованої реклами, Reddit не потребує великого обсягу чутливої інформації від своїх користувачів. Основний фокус Reddit — текстовий контент, тому платформа дозволяє користувачам залишатися анонімними, що значно знижує ризик порушення приватності під час аналізу даних. Такий підхід є важливим аспектом для автоматизованого збору та аналізу інформації, адже він мінімізує ймовірність обробки чутливих даних та забезпечує дотримання прав на конфіденційність.

Використання цієї платформи для дослідження негативних соціальних явищ, таких як мова ненависті чи кібербулінг, стає безпечнішим з погляду етичних питань, адже ідентифікація конкретних осіб залишається неможливою. Це створює умови для масштабованого аналізу публічних коментарів без ризику порушення приватності, що робить Reddit корисним інструментом для подібних досліджень і систем автоматизованого аналізу контенту.

4.1.4 Гнучкість та доступність API

Reddit має детально розроблену документацію для свого API, що дозволяє легко інтегрувати його у різні системи для збору та аналізу даних. Однією з головних переваг є можливість адаптації API до конкретних потреб проєкту, що робить його надзвичайно гнучким інструментом для розробників.

Reddit API також підтримує безперервний доступ до великого масиву даних у реальному часі, що особливо корисно для автоматизованого збору інформації для досліджень. На відміну від деяких інших платформ, API Reddit не має жорстких обмежень на обсяг запитів, що значно полегшує масштабовані дослідження або проекти, які вимагають великого обсягу даних для обробки. Reddit API також дозволяє отримувати різні типи контенту, включаючи пости, коментарі та метадані, що забезпечує комплексний підхід до збору даних

4.1.5 Проблеми з іншими соціальними мережами

Вибір Reddit також обґрунтований порівнянням з іншими популярними соціальними мережами:

Рисунок 4.3 – Порівняння з іншими соціальними мережами

У процесі створення інформаційної системи на основі даних із Facebook API було зіткнено з безпековими обмеженнями платформи. Зокрема, Facebook блокує можливість внесення змін у профіль або реєстрацію нових облікових записів розробника, якщо користувач входить із пристрою, який не був раніше використаний. Це пояснюється спробами захисту облікових записів від потенційно несанкціонованого доступу. Така практика є корисною для безпеки даних, але може затримувати процес розробки, особливо коли команді необхідно працювати з декількома пристроями.

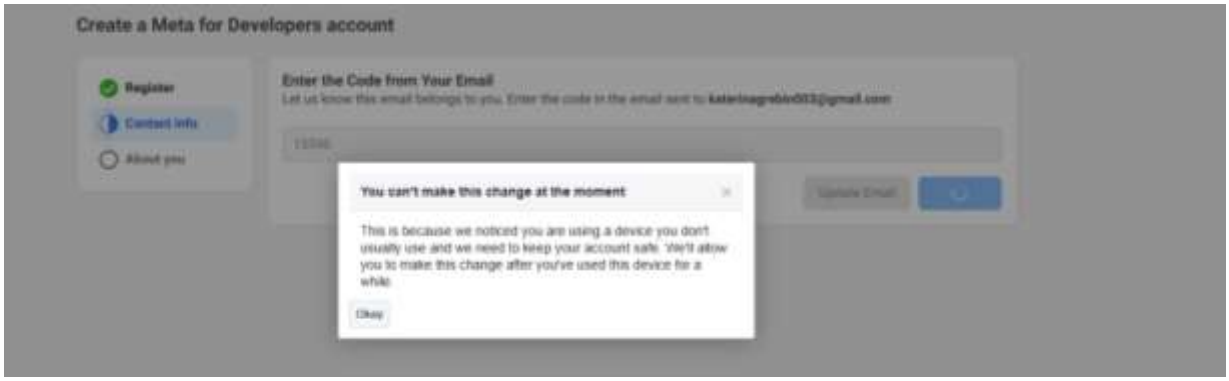

Рисунок 4.4 – Помилка при реєструванні в Meta for Developers

Також важливо зазначити, що отримання доступу до API Facebook у якості розробника може бути складним через кілька причин:

- суворі політики безпеки та конфіденційності: після скандалів з витоками даних, Facebook значно посилив вимоги до розробників. Всі додатки, які взаємодіють з API, повинні проходити ретельну перевірку, а деякі API запити вимагають затвердження політики використання даних;
- процес верифікації додатків: для доступу до більш розширених функцій API необхідно пройти процес верифікації вашого додатку. Це може включати надання додаткової інформації про вашу організацію, мету використання даних та виконання інших вимог, наприклад, підтвердження домену;
- обмеження на дані користувачів: Facebook має жорсткі обмеження на доступ до особистих даних користувачів. Щоб отримати доступ до певних типів даних, необхідно мати чітко обґрунтовану причину, яка відповідатиме політиці конфіденційності та вимогам платформи;
- регулярні зміни в API: Facebook постійно оновлює свої API, змінюючи або повністю припиняючи доступ до певних функцій, що ускладнює довгострокове планування проектів, які використовують їх API.

Reddit є платформою, яка відома своєю великою кількістю тематичних спільнот (сабреддітів), що дає змогу здійснювати глибокий тематичний аналіз різних аспектів суспільних і соціальних проблем. На Reddit часто можна знайти дискусії, що стосуються негативних тенденцій, таких як образи, ненависть, кібербулінг та інші негативні явища. Це дозволяє точно настроїти інформаційну систему для пошуку саме тих видів контенту, які необхідно аналізувати.

Загалом, Reddit є найбільш підходящою платформою для реалізації автоматизованого аналізу текстового контенту з соціальних мереж. Його відкритість, анонімність користувачів, відсутність значної кількості чутливих даних і доступність API роблять його оптимальним вибором для збору даних, які потім будуть використані для виявлення негативних тенденцій. Вибір цієї мережі забезпечує етичні, правові та технічні переваги в порівнянні з іншими популярними

платформами, що значно спрощує процес реалізації системи автоматизованого аналізу контенту.

4.1.6 Використання Reddit у дослідженнях

У статті [36] аналізується обговорення, пов'язані з ментальним здоров'ям, зібрані з Reddit, зокрема із сабреддітів, присвячених психічним розладам, таким як депресія, тривожність тощо. Використовуючи методи аналізу тексту і машинного навчання, автори дослідження вивчали, як користувачі описують свій досвід, і виділяли ключові теми для оцінки їхнього стану. Reddit виявився особливо корисним через анонімність та вільну форму дискусій, що дозволило користувачам відверто говорити про свій стан без страху осуду.

Дослідження [37] вивчало поширення мови ненависті у великих соціальних мережах, таких як Reddit. Автори використовували сабреддіти, які мають репутацію місць для токсичних дискусій, та аналізували мову, що використовувалася у постах і коментарях. Дослідження дозволило ідентифікувати основні ознаки мови ненависті і запропонувати методи її автоматичного виявлення. Reddit був обраний через його тематичні спільноти та можливість доступу до великої кількості публічних даних через API.

Стаття «Exploring Public Opinion on Reddit: A Case Study on the COVID-19 Pandemic» містить реакції та думки користувачів Reddit під час пандемії COVID-19. Використовуючи дані з сабреддітів, присвячених здоров'ю та новинам, автори вивчали настрої людей щодо різних заходів протидії пандемії, вакцинації та державних політик. Аналіз показав, як з часом змінювалися настрої громадськості, і як Reddit став платформою для обговорення важливих соціальних і політичних питань під час кризи [38].

В «Gender Representation in Online Communities: A Study of Reddit» дослідники проаналізували, як гендерні ролі та питання рівності обговорюються на Reddit. Сабреддіти, присвячені питанням рівності, гендерних ролей і прав жінок, стали предметом вивчення. Аналіз показав відмінності в тому, як чоловіки і жінки

представляють себе в онлайн-дискусіях і як вони реагують на певні соціальні питання. Reddit став важливою платформою для збору великої кількості думок з широкого кола тем, що дозволило отримати репрезентативні дані [39].

А наприклад, в [40] проводиться аналіз поширення фейкових новин та дезінформації на Reddit. Автори зібрали дані з сабреддітів, що обговорюють новини, і визначили джерела фейкових новин, вивчали їх вплив на користувачів та механізми їх поширення. Дослідження показало, як новини можуть маніпулювати громадською думкою та створювати інформаційні бульбашки. Reddit надав можливість для аналізу дезінформації в реальному часі через доступ до коментарів і обговорень новин.

Отже, всі ці дослідження демонструють, що Reddit є потужним інструментом для аналізу соціальних явищ, оскільки платформа надає широкий доступ до текстового контенту, дозволяє вивчати теми, що стосуються негативних тенденцій, а також сприяє збору репрезентативних даних. Завдяки доступності API, анонімності користувачів та багатогранності тем, Reddit стає ефективною базою для проведення різноманітних соціальних та психологічних досліджень.

4.2 Структура інформаційної системи

Інформаційна система, розроблена для аналізу контенту в соціальних мережах, зокрема для виявлення негативних трендів, таких як мова ненависті, складається з кількох ключових компонентів, кожен з яких виконує свою функцію у процесі збору, обробки, аналізу та візуалізації даних. Структура системи забезпечує автоматизований процес виявлення негативних явищ в онлайн-дискусіях та надає зручні інструменти для користувачів для фільтрації та візуалізації результатів. Система включає кілька основних компонентів, які забезпечують автоматизований збір даних, їх обробку та візуалізацію результатів.

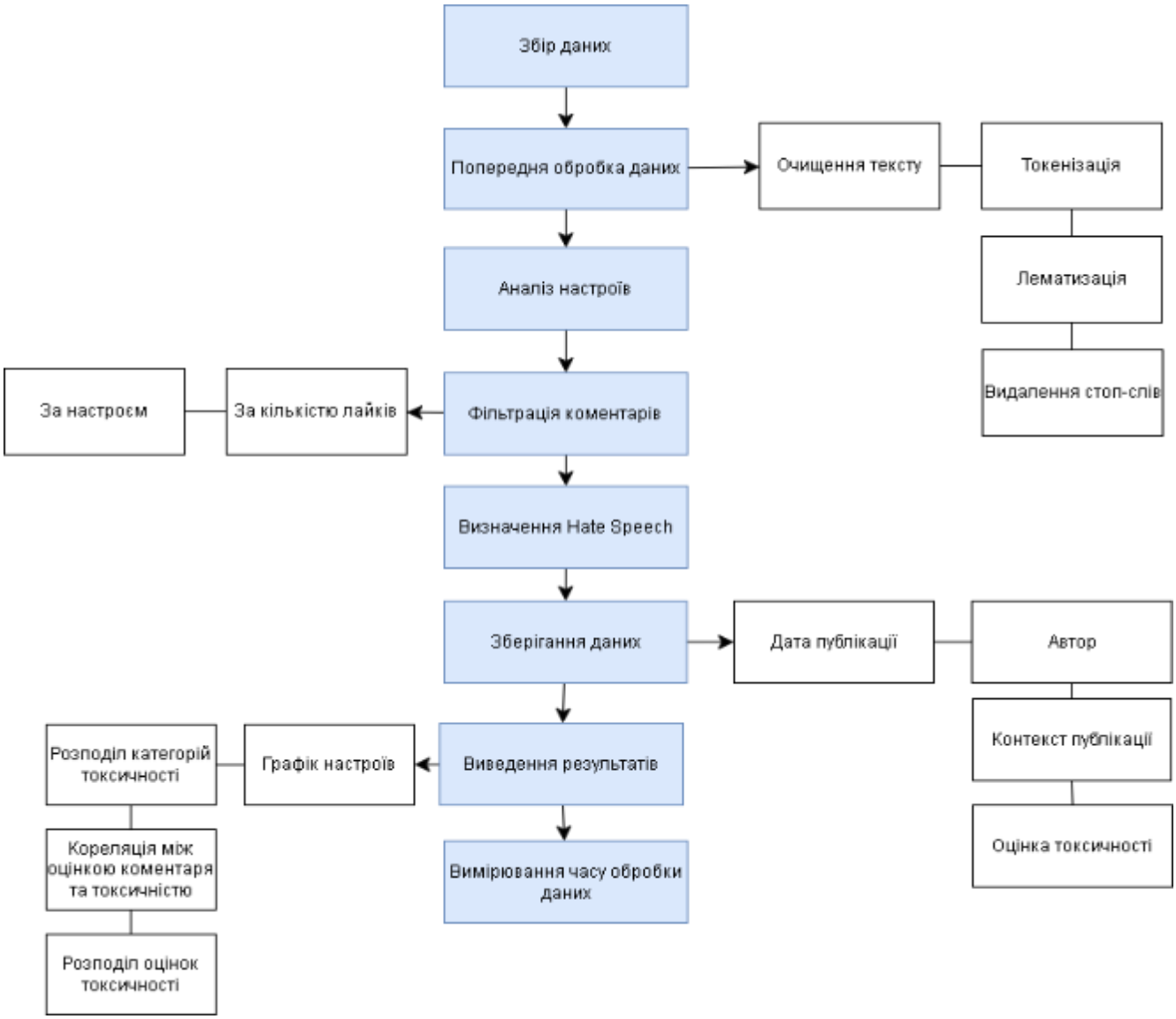

Рисунок 4.5 – Структура інформаційної системи

Основні компоненти системи:

– збір даних: за допомогою API платформи Reddit здійснюється отримання текстових даних з популярних публікацій у вибраних субреддітах;

– аналіз настроїв: використовуючи бібліотеку VADER Sentiment Analyzer, тексти класифікуються на позитивні, негативні та нейтральні;

– фільтрація даних: користувач має можливість фільтрувати коментарі за мінімальною кількістю лайків та настроєм;

– визначення hate speech: система також дозволяє виявляти мову ненависті в коментарях за допомогою спеціальних моделей;

– візуалізація результатів: результати аналізу настроїв виводяться у вигляді графіків, що відображають розподіл настроїв у реальному часі;

– зберігання даних: тільки коментарі, що були класифіковані як мова ненависті, зберігаються в базі даних разом з їхніми метаданими, включаючи дату публікації, автора та контекст. Це дозволяє відстежувати динаміку появи ненависницького контенту, створювати аналітичні звіти, ідентифікувати проблемні зони на платформі, де найчастіше виникає токсичний контент;

– вимірювання часу обробки даних: система дозволяє відслідковувати час обробки даних на кожному етапі, що допомагає оптимізувати продуктивність системи. Інформація про час обробки відображається у дебаг-вікні.

Ці компоненти утворюють основний ланцюг процесів, який дозволяє системі зібрати необхідні дані, проаналізувати їх, відфільтрувати та візуалізувати результати для подальшого аналізу.

Одним з перших етапів є збір даних з платформ, таких як Reddit, для аналізу коментарів на популярні публікації. Для цього використовується API Reddit, який дозволяє зібрати коментарі з вибраних субреддітів. Це дозволяє отримати великий обсяг коментарів для подальшого аналізу. Для збору даних використовуємо API Reddit через бібліотеку PRAW (Python Reddit API Wrapper), яка дозволяє здійснювати з'єднання з платформою і отримувати публікації з обраних субреддітів. Публікації вибираються з категорії hot, що означає найпопулярніші і найбільш актуальні пости в певний момент часу.

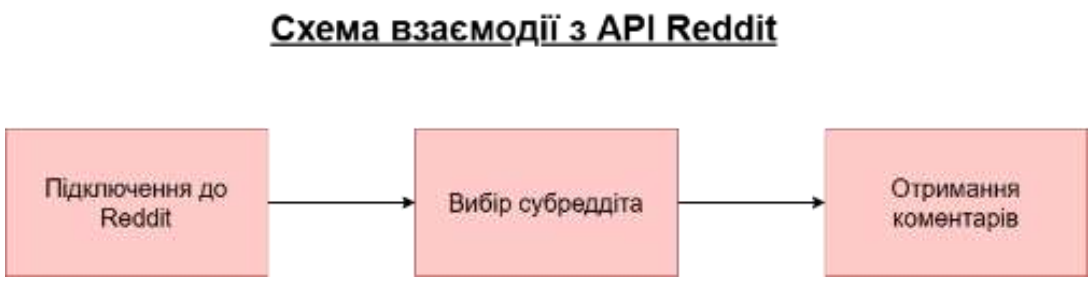

Рисунок 4.6 – Отримання даних через PRAW

Етапи збору даних подані нижче.

Ініціалізація з'єднання з Reddit. Встановлення з'єднання через API Reddit з використанням авторизаційних ключів (Client ID, Client Secret, User-Agent).

Рисунок 4.7 – Підключення до Reddit

Отримання публікацій з субреддітів. Система збирає публікації з найбільш актуальних категорій, таких як hot, що дозволяє отримати найбільш обговорювані пости.

Обмеження кількості коментарів. Щоб уникнути перевантаження системи великою кількістю даних, що може негативно вплинути на продуктивність та час обробки, запроваджується обмеження на кількість коментарів, які збираються з кожної публікації. Для кожного посту вибирається не більше 10 коментарів, що дозволяє оптимізувати процес збору даних та запобігти надмірному споживанню ресурсів. Це також допомагає уникнути зайвого дублювання інформації, особливо у випадках, коли публікації містять тисячі коментарів, багато з яких можуть бути повторюваними або не нести суттєвого значення для аналізу.

```
def collect_comments(api_id, api_secret, user_agent, subreddit_name, post_limit):
    try:
        reddit = praw.Reddit(client_id=api_id, client_secret=api_secret, user_agent=user_agent)
        subreddit = reddit.subreddit(subreddit_name)
        comments = []

        for submission in subreddit.hot(limit=post_limit):
            submission.comments.replace_more(limit=0)
            count = 0
            for comment in submission.comments.list():
                if count >= 10:
                    break
                comments.append({'text': comment.body, 'score': comment.score})
                count += 1
```

Рисунок 4.8 – Збір коментарів

Цей етап є необхідним для подальшої обробки і аналізу текстових даних.

Збір даних є першим етапом у процесі аналізу контенту, оскільки без коректного збору даних неможливо реалізувати наступні етапи обробки та аналізу.

Однією з ключових особливостей цієї інформаційної системи є можливість вибору методу аналізу настроїв. Користувач може вибрати один із кількох методів для оцінки емоційного забарвлення коментарів, що дає змогу адаптувати систему під конкретні потреби аналізу та обробки тексту. Система підтримує такі методи:

– VADER Sentiment Analysis .Це інструмент, орієнтований на аналіз коротких текстів, таких як коментарі на платформах типу Reddit. VADER класифікує текст на три категорії: позитивний, негативний, нейтральний, і використовує compound score, який є комплексною оцінкою емоційного забарвлення тексту. Якщо compound score більше 0.05, коментар вважається позитивним; якщо менше -0.05 – негативним; у проміжному діапазоні – нейтральним. Цей метод є швидким і ефективним для роботи з великою кількістю коротких коментарів, що є типовими для соціальних мереж;

```python
def analyze_sentiment_vader(comments):
    sentiments = []
    for comment in comments:
        analysis = vader_analyzer.polarity_scores(comment['text'])
        if analysis['compound'] >= 0.05:
            sentiments.append('Positive')
            sentiment_counts['Positive'] += 1
        elif analysis['compound'] <= -0.05:
            sentiments.append('Negative')
            sentiment_counts['Negative'] += 1
        else:
            sentiments.append('Neutral')
            sentiment_counts['Neutral'] += 1
        sentiment_values.append(sentiments[-1])
    return sentiments
```

Рисунок 4.9 – Код для аналізу настроїв за допомогою VADER

– TextBlob. Цей інструмент працює на основі простого аналізу полярності тексту, де результати можуть бути відображені в значеннях від -1 (негативний) до 1 (позитивний). Для аналізу більш складних або довших текстів, де контекст може

бути важливим, TextBlob є корисним завдяки своїй здатності оцінювати полярність більш гнучким методом, що дає можливість здійснювати більш тонкий аналіз;

```
72
73  def analyze_sentiment_textblob(comments):
74      sentiments = []
75      for comment in comments:
76          analysis = TextBlob(comment['text']).sentiment.polarity
77          if analysis > 0:
78              sentiments.append('Positive')
79              sentiment_counts['Positive'] += 1
80          elif analysis < 0:
81              sentiments.append('Negative')
82              sentiment_counts['Negative'] += 1
83          else:
84              sentiments.append('Neutral')
85              sentiment_counts['Neutral'] += 1
86          sentiment_values.append(sentiments[-1])
87      return sentiments
88
```

Рисунок 4.10 – Код для аналізу настроїв за допомогою TextBlob

– BERT Sentiment Analysis. Модель BERT (Bidirectional Encoder Representations from Transformers) використовує методи глибокого навчання для оцінки настроїв в текстах з урахуванням контексту. Вона забезпечує високу точність аналізу, особливо для складних текстів з багатозначним контекстом або сарказмом. DistilBERT – легша версія цієї моделі, яка також може бути використана для швидшої обробки тексту з високою точністю.

```
91  def analyze_sentiment_bert(comments):
92      sentiments = []
93      for comment in comments:
94          analysis = sentiment_pipeline(comment['text'])[0]
95          if analysis['label'] == 'POSITIVE':
96              sentiments.append('Positive')
97              sentiment_counts['Positive'] += 1
98          elif analysis['label'] == 'NEGATIVE':
99              sentiments.append('Negative')
100             sentiment_counts['Negative'] += 1
101         else:
102             sentiments.append('Neutral')
103             sentiment_counts['Neutral'] += 1
104         sentiment_values.append(sentiments[-1])
105     return sentiments
106
```

Рисунок 4.11 – Код для аналізу настроїв за допомогою BERT

Аналіз настроїв дозволяє виділяти коментарі, що можуть містити негативні елементи, такі як мова ненависті або агресивні висловлювання.

Інструмент використовує compound score, що є комплексною оцінкою настрою тексту. Якщо score більше за 0.05, коментар вважається позитивним; якщо менше за -0.05 – негативним; в іншому випадку – нейтральним.

Фільтрація даних дозволяє вибирати лише ті коментарі, які відповідають певним критеріям.

Користувач може фільтрувати коментарі за мінімальною кількістю лайків, код представлено на рисунку:

```
def filter_by_likes(comments, min_likes):
    return [comment for comment in comments if comment['score'] >= min_likes]
```

Рисунок 4.12 – Фільтрація за кількістю лайків

А також можна фільтрувати коментарі за настроєм, вибираючи лише позитивні, негативні або нейтральні коментарі для подальшого аналізу:

```
def filter_by_sentiment(comments, sentiments, selected_sentiment):
    if selected_sentiment == "All":
        return comments, sentiments
    filtered_comments = []
    filtered_sentiments = []
    for comment, sentiment in zip(comments, sentiments):
        if sentiment == selected_sentiment:
            filtered_comments.append(comment)
            filtered_sentiments.append(sentiment)
    return filtered_comments, filtered_sentiments
```

Рисунок 4.13 – Фільтрація за настроєм

Крім базового аналізу настроїв, система також дозволяє користувачам визначати наявність hate speech у коментарях. Для цього використовується спеціально навчена модель Toxic BERT, яка орієнтована на виявлення токсичних виразів у текстах.

Модель Toxic BERT реалізує pipeline для класифікації коментарів на категорії toxic (токсичні) та non-toxic (нетоксичні). Користувач може вказати поріг токсичності (наприклад, score ≥ 0.5), щоб відфільтрувати лише найбільш токсичні коментарі.

```
def detect_hate_speech():
    try:
        api_id = api_id_var.get()
        api_secret = api_secret_var.get()
        user_agent = user_agent_var.get()
        subreddit_name = subreddit_var.get()
        post_limit = int(post_limit_var.get())

        if not api_id or not api_secret or not user_agent or not subreddit_name:
            error_message.set("Please fill in all the fields!")
            return

        if post_limit <= 0:
            error_message.set("Post limit must be a positive integer.")
            return

        comments = collect_comments(api_id, api_secret, user_agent, subreddit_name, post_limit)
        num_comments = len(comments)
        comment_count_label.config(text=f"Зібрано коментарів: {num_comments}")

        if num_comments == 0:
            error_message.set("No comments collected. Please check the subreddit or API credentials.")
            return

        # Аналіз hate speech
        hate_speech_results = analyze_hate_speech(comments)

        # Відобразити результати
        comment_display.delete(1.0, END)
        if hate_speech_results:
            for comment, label, score in hate_speech_results:
                comment_display.insert(
                    END, f"Hate Speech Detected!\nScore: {score:.2f}\nLabel: {label}\nText: {comment['text']}\n\n"+"-"*50+"\n"
                )
        else:
            comment_display.insert(END, "No hate speech detected.\n")

        error_message.set("")
    except ValueError:
        error_message.set("Invalid value entered. Please check your inputs.")
```

Рисунок 4.14 – Функція для виявлення hate speech

Це важливий інструмент для модерації контенту в соціальних мережах, оскільки він дає змогу автоматично виявляти образливі або неприязні висловлювання, що можуть спричиняти конфлікти серед користувачів або порушувати правила поведінки.

Після виявлення hate speech, програма може виконувати кілька важливих етапів для подальшої обробки, зберігання та аналізу даних. Ці етапи дозволяють не тільки класифікувати та фільтрувати токсичні коментарі, але й використовувати отриману інформацію для довгострокового моніторингу та аналізу негативних тенденцій у соціальних мережах.

Після класифікації коментарів як токсичних або таких, що містять мову ненависті, система зберігає їх у базі даних разом з метаданими. Це забезпечує можливість подальшого аналізу й створення звітів для оцінки динаміки й поширення токсичного контенту. Метадані можуть включати:

– дата публікації коментаря;

– ідентифікатор та ім'я автора коментаря;

– контекст, у якому був опублікований коментар (наприклад, тема або категорія посту).

Це надасть можливість:

– вивчати динаміку появи ненависницького контенту, тобто можна відстежувати, як і коли токсичні коментарі стають частіше або рідше, та ідентифікувати загальні тенденції поширення негативних висловлювань;

– створювати аналітичні звіти, які допоможуть модераторам і адміністраторам платформ приймати рішення щодо політики щодо ненависті ;

– виявляти найбільш проблемні ділянки платформи, де найчастіше виникають конфлікти або ненависть.

Для візуалізації результатів аналізу настроїв використовується бібліотека matplotlib. Графік розподілу настроїв оновлюється в реальному часі, що дає змогу користувачеві спостерігати за змінами в настроях серед коментарів по ходу збору та аналізу даних.Графік показує відсотковий розподіл позитивних, негативних і нейтральних коментарів, що дозволяє отримати наочне уявлення про емоційне забарвлення обговорень.

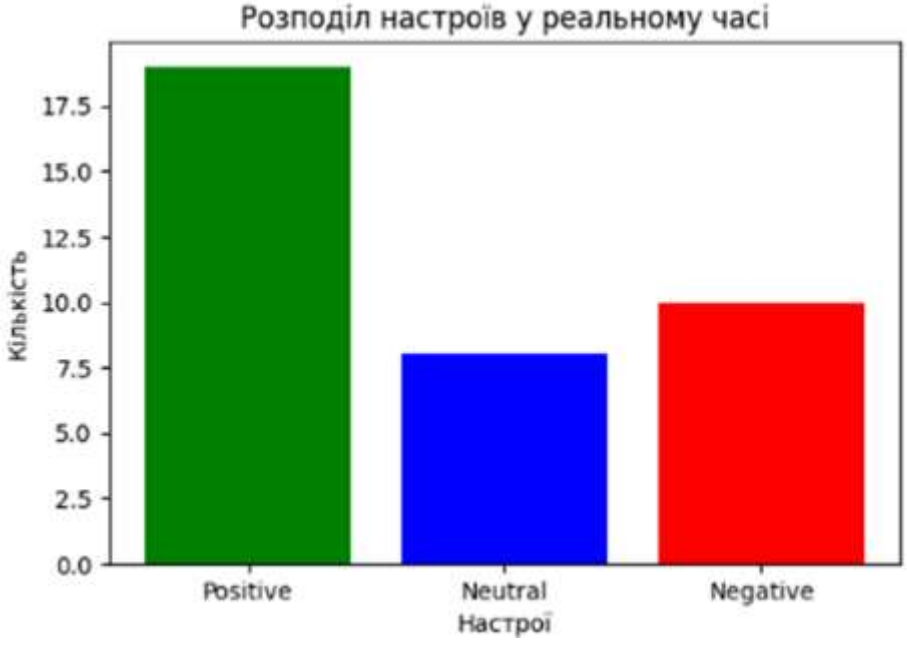

Рисунок 4.15 – Графік розподілу настроїв

Результати аналізу коментарів з hate speech можуть бути візуалізовані у вигляді різних графіків і діаграм, що допоможуть зручніше інтерпретувати отримані результати. Такі графіки дозволяють зрозуміти масштаб проблеми і виявляти негативні тренди, наприклад, як токсичний контент розподіляється в часі або серед користувачів. Основні види візуалізації описані нижче.

Розподіл категорій токсичності. Цей графік показує частки коментарів з високою та помірною токсичністю. Він допомагає розуміти, наскільки серйозним є токсичний контент у певний момент часу.

Кореляція між рейтингом коментаря (Comment Score) і токсичністю. Графік дозволяє побачити взаємозв'язок між популярністю коментаря (рейтингом) і рівнем токсичності. Це може вказати на тенденції у взаємодії користувачів із негативним контентом.

Розподіл індексів токсичності. Відображає кількість коментарів із різними значеннями індексу токсичності, що допомагає виявити найтоксичніші сегменти контенту.

Ці візуалізації використовуються для створення реального аналітичного уявлення про поширеність негативних явищ і можуть стати основою для рішень щодо модерації платформи.

4.3 Тестування та аналіз результатів програми

Важливою частиною системи є тестування на основі реальних даних та аналіз отриманих результатів. Програма, розроблена для збору та аналізу коментарів із соціальних мереж, дозволяє ефективно фільтрувати й обробляти дані, щоб забезпечити точний аналіз настроїв та виявлення важливих коментарів. Одна з ключових можливостей програми – це фільтрація коментарів за кількістю лайків.

Наприклад, при тому, що зібрано 39 коментарів, на екран виведено тільки один коментар, з кількістю лайків 13. Це пояснюється тим, що в програмі реалізовано мінімальний поріг лайків для виведення коментарів, що складає 10

лайків. Тільки ті коментарі, що набрали більше або дорівнюють 10 лайкам, будуть показані на екрані для подальшого аналізу.

Рисунок 4.16 – Фільтрація за кількістю лайків

Фільтрація за кількістю лайків ілюструє цей процес, демонструючи, як коментарі з меншою кількістю лайків не потрапляють у кінцевий список, що дозволяє акцентувати увагу лише на найбільш популярних постах. Такий підхід дозволяє зосередити аналіз на найбільш значущих для аудиторії коментарях, що мають більший вплив.

Ще однією важливою функцією є фільтрація коментарів за настроєм. Програма дає можливість вибирати коментарі, що мають певний настрій: позитивний, негативний або нейтральний. Це дозволяє ефективно відокремити важливі коментарі, що містять сильні емоційні реакції, від менш значущих. Такі фільтри дозволяють отримати більш точне уявлення про емоційну атмосферу обговорення.

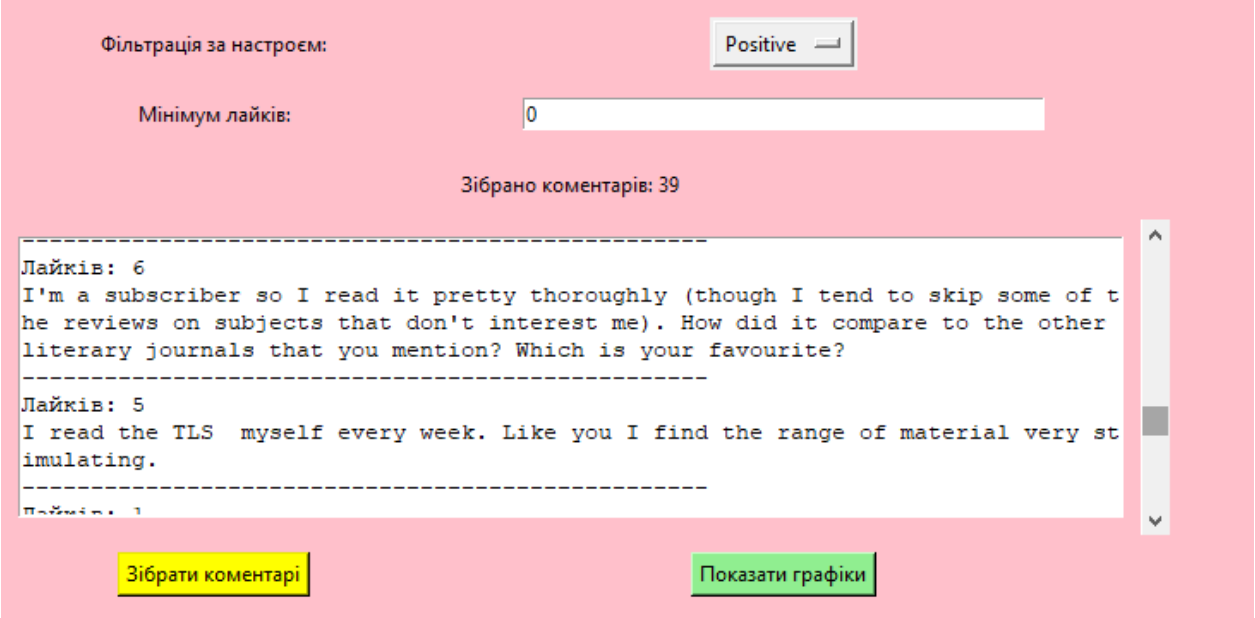

Рисунок 4.17 – Фільтрація за настроєм

Фільтрація за настроєм показує, як користувач може фільтрувати коментарі за їх емоційною забарвленістю, що допомагає зосередитися на коментарях з виразним настроєм, будь то позитивним або негативним. Це значно покращує якість аналізу, дозволяючи точніше визначати основні тенденції та реакції учасників обговорення.

Система також оснащена механізмами для виявлення hate speech, що дає змогу автоматично виявляти коментарі, які порушують норми поведінки, містять образи або дискримінаційні висловлювання. Це важлива частина аналізу, оскільки вона допомагає забезпечити безпеку онлайн-простору та попереджати поширення ненависті. Виявлення hate speech демонструє, як програма може автоматично позначати коментарі, що містять ненависницьку мову. Це дозволяє оперативно реагувати на токсичні обговорення і зберігати конструктивну атмосферу.

Рисунок 4.18 – Виявлення hate speech

Виявлення hate speech в програмі реалізовано за допомогою автоматичних алгоритмів, що аналізують текст коментарів на наявність образливих, дискримінаційних або агресивних висловлювань. Цей процес ґрунтується на кількох основних підходах:

– аналіз ключових слів: програма використовує базу даних із забороненими або потенційно шкідливими словами, фразами та виразами, які можуть вказувати на мову ненависті. Вона порівнює коментарі з цими забороненими термінами і маркує їх, якщо вони містять підозрілі елементи. Наприклад, програма може виявити слова, що закликають до насильства, расової чи релігійної нетерпимості, дискримінації за статтю чи сексуальною орієнтацією;

– контекстуальний аналіз: для точнішого виявлення hate speech програма використовує методи контекстного аналізу, щоб не лише перевіряти наявність окремих слів, але й аналізувати їх контекст. Наприклад, фрази, що можуть здаватися нейтральними в окремому контексті, але набувають агресивного підтексту в поєднанні з іншими словами;

– аналіз емоцій: оскільки hate speech часто супроводжується сильними негативними емоціями, програма використовує аналіз настроїв для виявлення коментарів, що мають виразно агресивний або ворожий характер. Наприклад, коментарі, що містять надмірно негативний або токсичний емоційний контекст, можуть бути позначені як такі, що ймовірно містять hate speech;

– штучний інтелект і обробка природної мови (NLP): програма застосовує методи обробки природної мови для глибокого аналізу структури та змісту коментарів.

Виявлення hate speech демонструє, як програма позначає коментарі, що містять ненависницьку мову, використовуючи вищезгадані методи. Під час процесу аналізу всі зібрані коментарі зберігаються в базі даних разом із метаданими, такими як дата публікації, автор та інші характеристики. Це дозволяє не лише проводити детальний аналіз поточних даних, але й виконувати ретроспективне дослідження, аналізуючи динаміку змін у поведінці користувачів. Збереження коментарів також надає можливість повторного використання даних для навчання, що допомагає підвищувати точність виявлення негативних явищ, таких як мова ненависті чи токсичність.

	id	text	author	subreddit	score	timestamp	toxicity_label	toxicity_score
1	1	He's a real G. Respect. …	TheDangerdog	4chan	1	2024-12-07T17:56:19.059576	toxic	0.5689394593323853
2	2	Love this. Fuck yeah.	equatOpotamus	4chan	1	2024-12-07T17:56:19.554316	toxic	0.9742637787269592
3	3	I like chubby women. I thought she …	ceciliforester	4chan	1	2024-12-07T17:56:19.924326	toxic	0.8279280411114044
4	4	IDGAF what anyone says; his wife wa…	upyoursize	4chan	1	2024-12-07T17:56:20.346198	toxic	0.6001749367427826
5	5	Spot on. If you're the type of …	beefequinta	4chan	1	2024-12-07T17:56:21.092204	toxic	0.9883595970503235
6	6	Like I said, a lot of men are …	MentalRadish3490	4chan	1	2024-12-07T17:56:21.311617	toxic	0.5731522946753293
7	7	i mean thats what you are dude lol…	thatemelkoun	4chan	1	2024-12-07T17:56:21.755423	toxic	0.5905435005829663
8	8	Its common known fact that men are …	PooeyPatoeei	4chan	1	2024-12-07T17:56:21.824347	toxic	0.6204621195793315
9	9	BROOO HOOOOOO I JUST FOUND OUT FUCK	Suitable-Quiet5683	4chan	1	2024-12-07T17:56:22.175809	toxic	0.9880758523941040
10	10	That is a fucked up and most …	theeldergodi	Unexpected	1	2024-12-07T17:56:44.715822	toxic	0.9933099656295776
11	11	They're pretty energy efficient …	Doyoucondemnhummus	Unexpected	3	2024-12-07T17:56:46.853108	toxic	0.8710183501243591
12	12	Don't forget to do your stretches …	RissaCrochets	Unexpected	1	2024-12-07T17:56:47.016670	toxic	0.6490002409935
13	13	He fucking nailed that landing lol	ScrantonDangler	Unexpected	247	2024-12-07T17:56:47.279965	toxic	0.9886566400527954
14	14	Title made me expect porn	OtherwiseTop2849	Unexpected	16	2024-12-07T17:56:47.430564	toxic	0.7355018854141235
15	15	Son of a bitch looking at his car …	DeathWings80	Unexpected	6	2024-12-07T17:56:47.683855	toxic	0.9947794675827026
16	16	That was a 10/10 Olympic-worthy …	ClaR6	Unexpected	2	2024-12-07T17:56:48.050905	toxic	0.9495906829833984
17	17	i got hit by a car some time ago …	HammerBgError404	Unexpected	36	2024-12-07T17:56:48.799902	toxic	0.9115064144134521
18	18	Probably the saddle might have got …	soccersonbounce	Unexpected	26	2024-12-07T17:56:49.358408	toxic	0.9418509602546692
19	19	Simply because the dumbass didn't …	ShiroGameOsu	Unexpected	1	2024-12-07T17:56:49.785266	toxic	0.5451943874359131
20	20	That's just because any title makes…	Apricot9742	Unexpected	1	2024-12-07T17:56:50.125357	toxic	0.6048573660850525
21	21	He is a prick.	True-Put-3712	Unexpected	0	2024-12-07T17:56:50.352750	toxic	0.9265033602714539
22	22	its an intersection you prick	evilmojoyousuck	Unexpected	9	2024-12-07T17:56:50.401619	toxic	0.6322257041931152

Рисунок 4.19 – База даних hate speech

Окрім того, для аналізу розподілу настроїв серед коментарів використовуються графіки та гістограми, що відображають зміни настроїв у реальному часі.

Гістограма «Розподіл настроїв у реальному часі» при методі VADER показує, як розподіляються позитивні, негативні та нейтральні коментарі протягом аналізу, використовуючи метод VADER для аналізу настроїв. Цей графік дозволяє користувачам швидко оцінити, який настрій переважає серед зібраних коментарів.

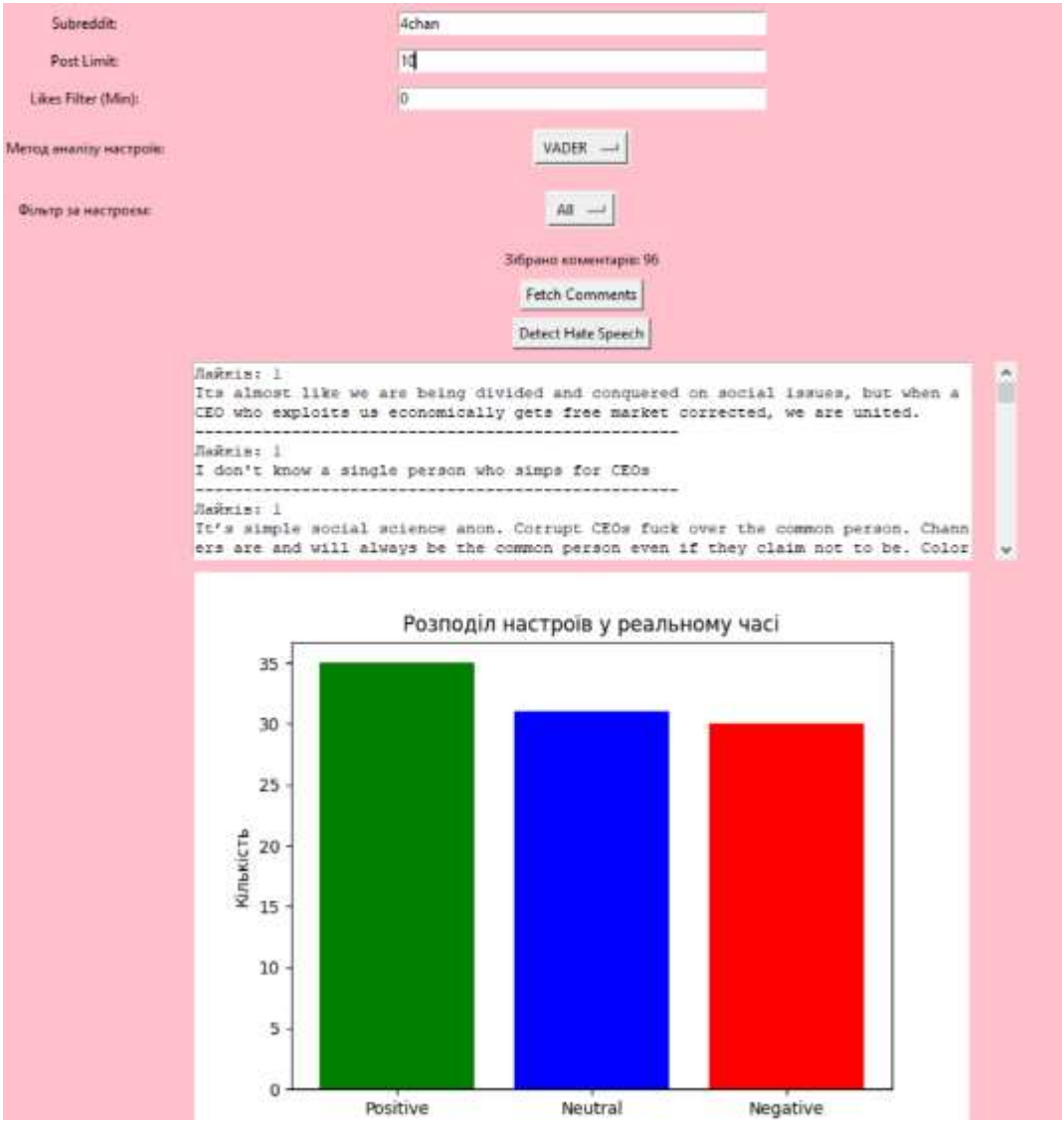

Рисунок 4.20 – Гістограма «Розподіл настроїв у реальному часі» при методі VADER

Гістограма «Розподіл настроїв у реальному часі» при методі TextBlob ілюструє, як інший метод, TextBlob, дає змогу досліджувати емоційну динаміку на основі інших алгоритмів обробки тексту, порівнюючи результати з VADER і допомагаючи точніше визначати настрій в коментарях.

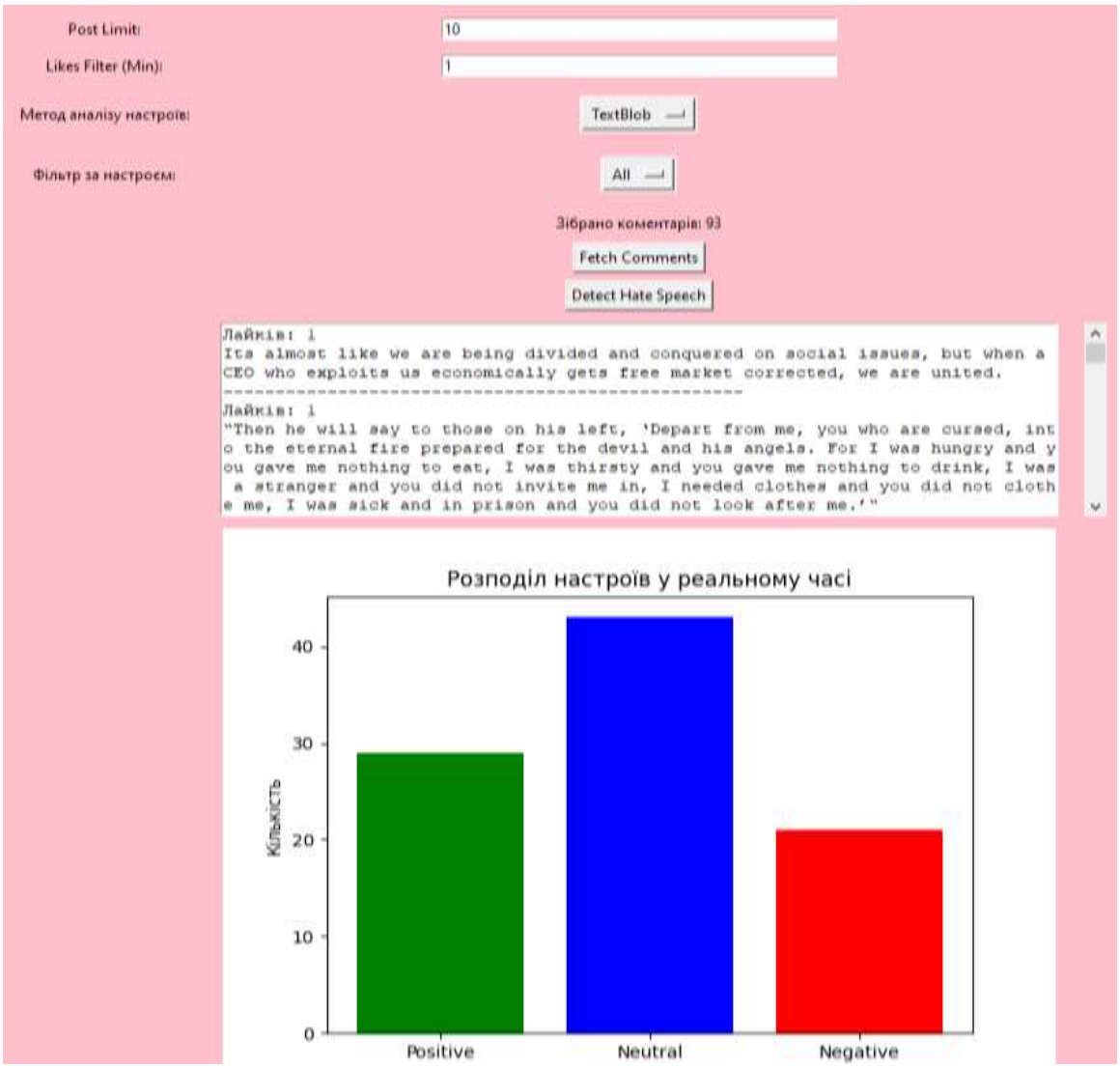

Рисунок 4.21 – Гістограма «Розподіл настроїв у реальному часі» при методі TextBlob

Гістограма «Розподіл настроїв у реальному часі» при методі BERT надає ще один погляд на емоційний розподіл коментарів, використовуючи більш складну модель BERT, яка дозволяє точніше оцінювати зміст коментарів завдяки глибшому розумінню контексту.

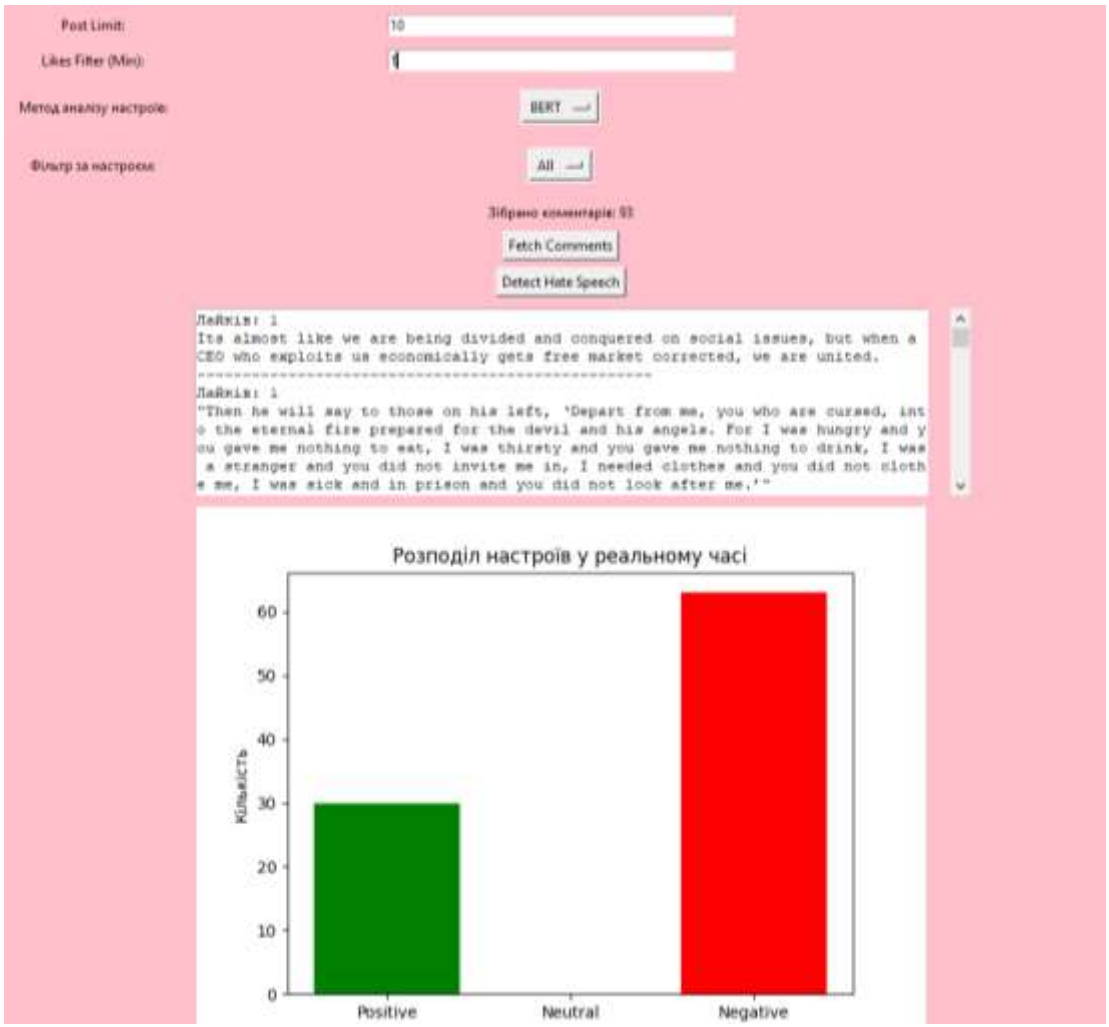

Рисунок 4.22 – Гістограма «Розподіл настроїв у реальному часі» при методі BERT

Ці графіки надають користувачу чітке уявлення про розподіл настроїв серед зібраних коментарів на будь-якому етапі збору і обробки даних, що дозволяє не лише оцінити поточний емоційний фон, а й виявити зміни в настрої за часом.

Під час тестування оцінюється ефективність та точність виявлення мови ненависті й надаються результати у вигляді графіків для візуалізації негативних явищ у коментарях.

Графік "Distribution of toxicity categories" (Розподіл категорій токсичності) дозволяє побачити, наскільки розповсюджені різні рівні токсичності серед проаналізованих коментарів. Коментарі класифікуються за рівнями, такими як

високий, середній або низький рівень токсичності, що дає змогу визначити загальну тенденцію токсичного контенту на платформі.

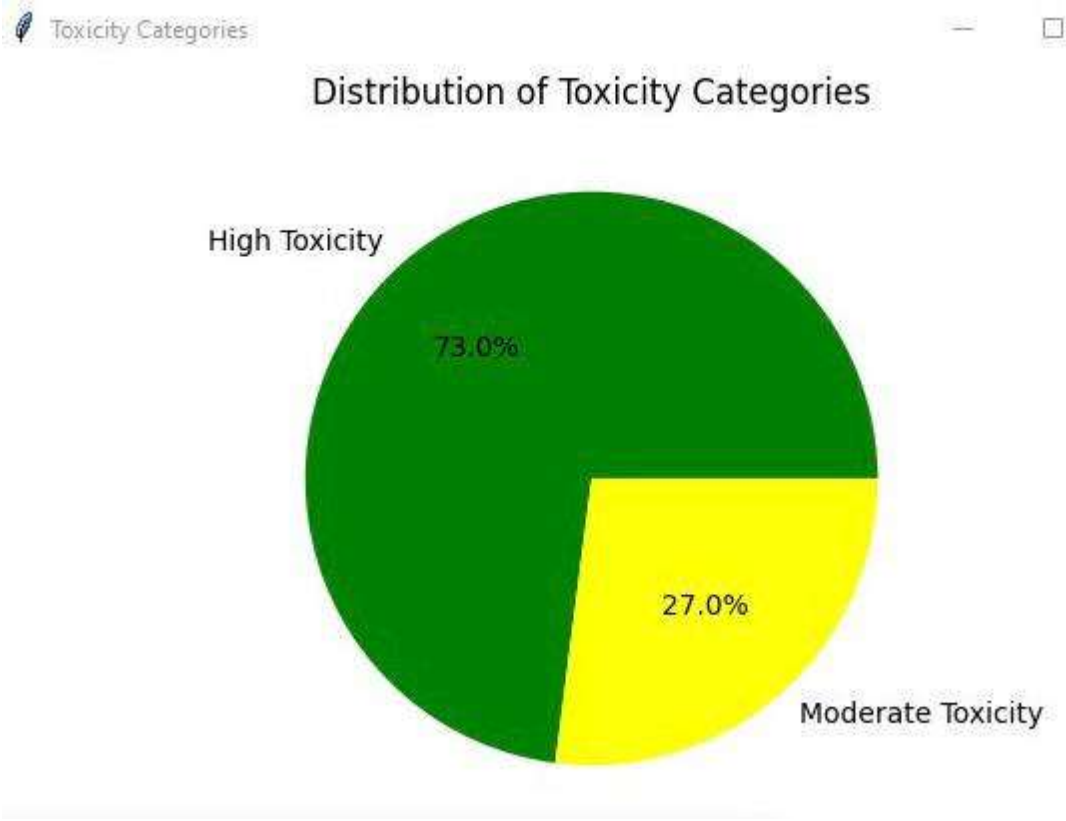

Рисунок 4.23 – Розподіл категорій токсичності

Завдяки цьому графіку можна побачити, який саме рівень токсичності переважає, що може вказувати на потенційні проблемні ділянки для модерації. Наприклад, якщо більшість коментарів мають середній рівень токсичності, то це може свідчити про необхідність посиленої модерації та вдосконалення автоматизованих фільтрів. Крім того, аналіз категорій токсичності може допомогти виявити, які користувачі чи групи є більш схильними до негативної поведінки.

Графік "Correlation between Comment Score and Toxicity" (Кореляція між рейтингом коментаря і токсичністю) відображає, як пов'язані популярність коментаря та його рівень токсичності. Рейтинг коментаря визначається кількістю лайків та дизлайків, що дає змогу оцінити рівень взаємодії користувачів з цим контентом.

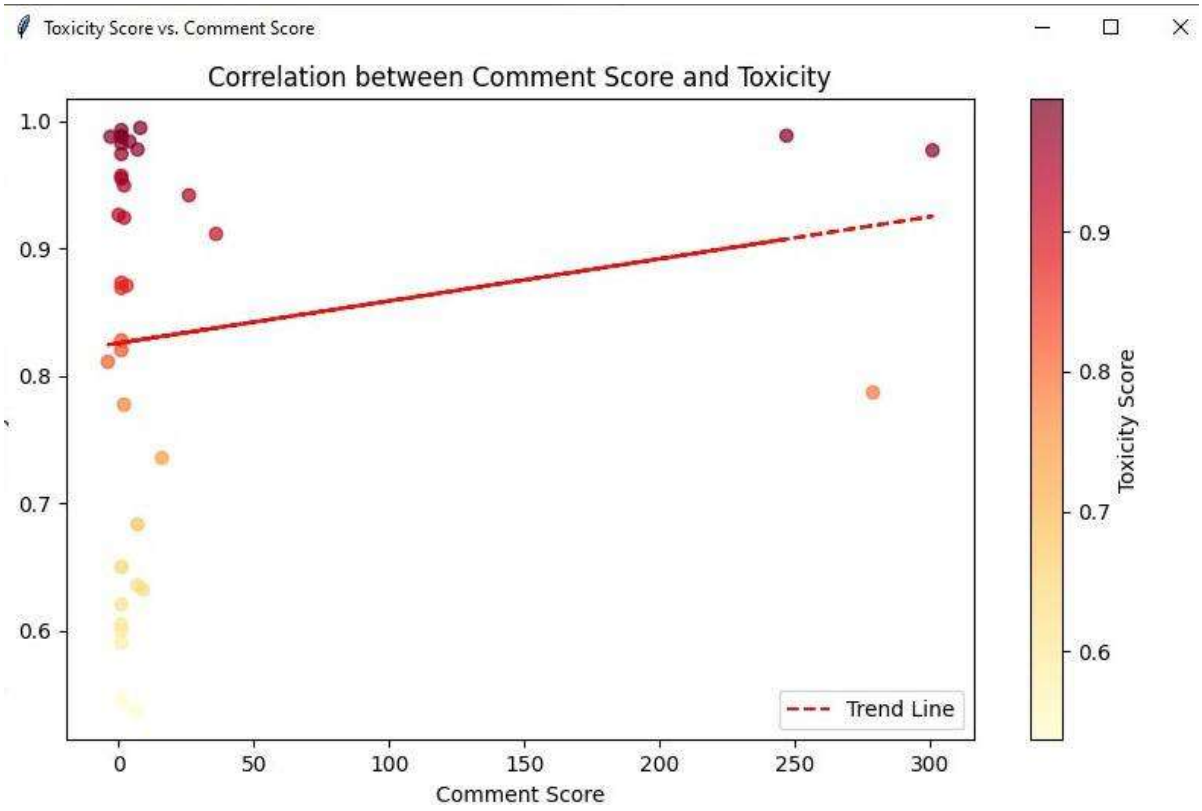

Рисунок 4.24 – Розподіл категорій токсичності

Важливим аспектом є виявлення того, чи популярні коментарі, що містять високий рівень токсичності, оскільки це може свідчити про проблеми з модерацією платформи. Якщо коментарі з високим рівнем токсичності отримують значну кількість позитивної взаємодії, це може вказувати на необхідність змін у політиці платформи щодо управління подібним контентом. Крім того, цей аналіз може показати, як сприймається токсичний контент користувачами, і допомогти ідентифікувати категорії коментарів, які найчастіше взаємодіють з негативним контентом.

Графік "Distribution of Toxicity Scores" (Розподіл індексів токсичності) показує, як рівень токсичності коментарів розподіляється по різних значеннях на платформі. Цей графік дозволяє оцінити загальний рівень токсичності контенту та виявити найбільш проблемні зони, де спостерігається висока концентрація токсичних коментарів. З його допомогою можна визначити, чи є певні періоди часу

або групи користувачів, що генерують більше токсичних повідомлень, що може вказувати на необхідність більш цілеспрямованої модерації..

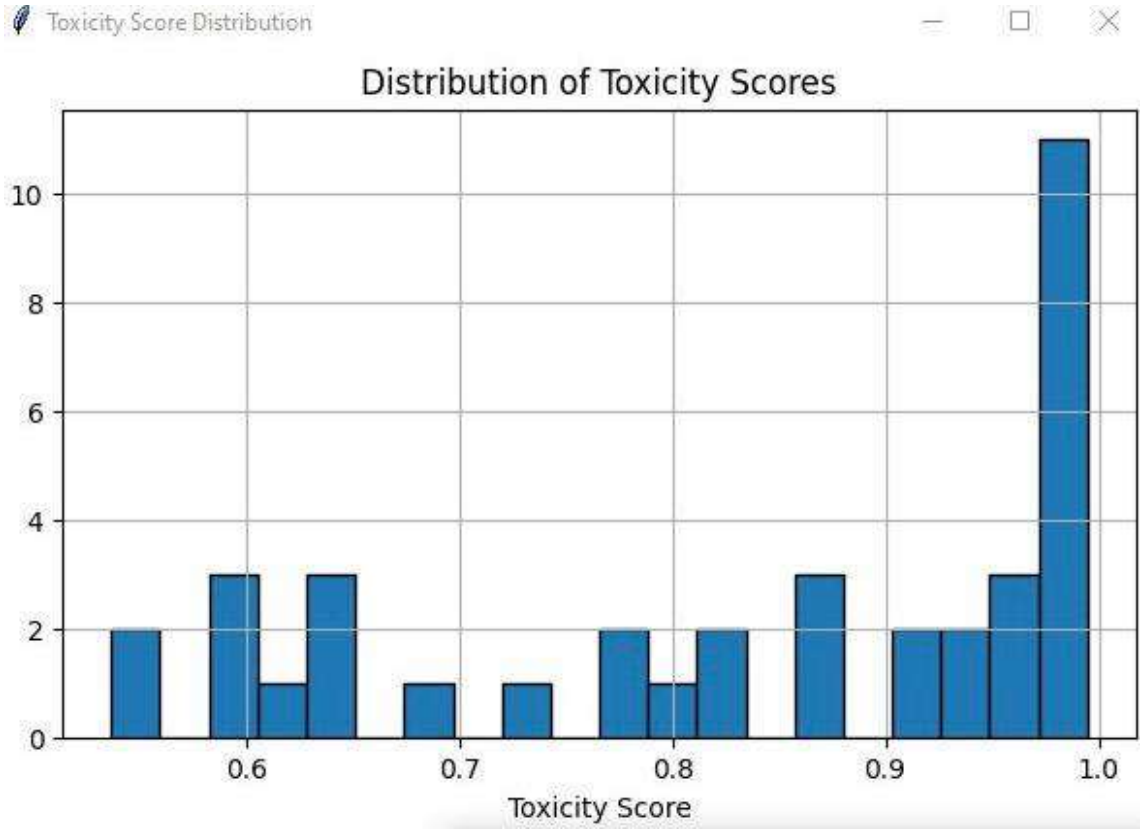

Рисунок 4.25 – Розподіл індексів токсичності

Загалом, графіки "Distribution of Toxicity Categories", "Correlation between Comment Score and Toxicity", та "Distribution of Toxicity Scores" забезпечують ефективне візуальне представлення даних, що дозволяє краще розуміти поведінку користувачів на платформі. Вони допомагають виявити ключові проблемні зони, такі як найбільш токсичні коментарі, їх взаємозв'язок з популярністю контенту та загальний рівень токсичності в коментарях. Ці графіки сприяють своєчасному реагуванню на негативні тенденції та оптимізації процесу модерації, що в свою чергу підвищує загальну безпеку та комфорт користувачів на платформі. Крім того, їх використання дозволяє здійснювати більш точне прогнозування токсичності в контенті та забезпечує основи для подальшого вдосконалення системи автоматизованого аналізу. Таким чином, ці візуалізації стають важливим інструментом для прийняття рішень і моніторингу на платформі.

Для кожного етапу обробки коментарів у системі ведеться ретельне вимірювання часу виконання. Це включає такі етапи, як збір даних, попередню обробку тексту, аналіз настроїв, а також процес виявлення мови ненависті. Вимірювання часу на кожному з етапів дозволяє виявити найбільш ресурсоємні процеси та оцінити загальну продуктивність системи.

Рисунок 4.26 – Вимірювання часу обробки даних

Дана інформація відображається в дебаг-вікні, і зрозуміти, де можуть бути затримки або можливості для оптимізації. Наприклад, якщо етап попередньої обробки даних займає надмірно багато часу, це може вказувати на потребу у вдосконаленні алгоритмів очищення або на необхідність обробки менших обсягів тексту за один раз. Вимірювання часу обробки є важливим елементом для оцінки та оптимізації роботи системи, зокрема у контексті масштабування при роботі з великими обсягами даних.

Оцінка точності моделі є критично важливим етапом для перевірки ефективності виявлення мови ненависті. Для цього порівнюються результати роботи запропонованої системи з результатами моделі Toxic BERT, яка доступна на платформі Hugging Face і є однією з провідних у розпізнаванні токсичних коментарів.

Для вимірювання точності використовуються метрики Precision, Recall та F1-score. Precision відображає точність системи, тобто скільки з передбачених токсичних коментарів дійсно є токсичними, Recall – здатність моделі виявляти всі токсичні коментарі, а F1-score є комбінованою метрикою, що об'єднує точність і повноту для оцінки загальної ефективності. У процесі тестування було виявлено, що коефіцієнти токсичності, отримані системою, значною мірою збігаються з результатами, отриманими за допомогою моделі Toxic BERT. Це свідчить про високий рівень точності запропонованої моделі в розпізнаванні токсичних коментарів.

Рисунок 4.27 – Оцінка ефективності

Точність і результативність, схожі на ті, що демонструє модель Toxic BERT, підтверджують ефективність використаних підходів і алгоритмів в рамках розробленої системи, що робить її надійною для реального застосування в автоматизованій модерації контенту.

Висновки до розділу 4

У розділі 4 було розглянуто процес проєктування та реалізації інформаційної системи для автоматизованого аналізу контенту в соціальних мережах.

Було обрано найбільш відповідну соціальну мережу для реалізації інформаційної системи. Після аналізу різних платформ, було вирішено використовувати Reddit через його відкритий доступ до даних та популярність серед цільової аудиторії. Ця платформа забезпечує широкий спектр контенту, що дозволяє здійснювати глибокий аналіз негативних тенденцій, таких як мова ненависті.

Структура системи була спроектована з урахуванням необхідності обробки великих обсягів даних та ефективного виявлення негативних явищ. Система включає модулі для збору даних, їх попередньої обробки, аналізу та збереження результатів. Завдяки модульності структура забезпечує гнучкість у розширенні та оптимізації процесів, що є важливим для подальшого вдосконалення системи.

Виконані тести показали, що система успішно реалізує поставлені завдання, зокрема аналізує тексти з високою точністю.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було успішно реалізовано комплексне дослідження методів аналізу контенту соціальних мереж та виявлення негативних тенденцій.

Важливою частиною роботи стало обгрунтування вибору методів і моделей для аналізу текстових даних. Було проаналізовано сучасні технології для обробки природної мови (NLP) та методи класифікації текстів. Серед розглянутих моделей булоособливо детально вивчено методи для виявлення мови ненависті.

У рамках проєктування інформаційної було вибрано платформу Reddit, яка надає доступ до текстовихданих через свій API , в цілому дозволило реалізувати систему без необхідності обробки великої кількості персональних даних користувачів.

В ході тестування інформаційної системи було проаналізовано її продуктивність, точність моделі, а також час обробки даних на різних етапах. Здійснено порівняння з відомою моделлю Toxic BERT, що підтвердило ефективність розробленого підходу до виявлення негативних явищ у соціальних мережах. Отримані результати демонструють високу точність розпізнавання негативного контенту та можливість використання системи для подальшого розвитку і впровадження в реальних умовах.

Загалом, результати даної роботи дозволяють зробити висновок про доцільність використання розробленої інформаційної системи для автоматизованого аналізу контенту в соціальних мережах, що сприятиме своєчасному виявленню та запобіганню поширенню негативних тенденцій в інтернет-просторі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Smith, J. "The Role of Social Networks in Modern Communication." *Journal of Media Studies*, 2021(date of access: 01.09.2024).

2. Zhong B. Social Media and Network Society. *Impact of Social Media on Communication and Business*. P. 5064–6073. URL: https://doi.org/10.4018/978-1-4666-9518-4.les7 (date of access: 02.09.2024).

3. Jyoti Suraj Harchekar. Impact of Social Media on Society. *International Journal of Engineering Research and*. 2017. Vol. V6, no. 07. URL: https://doi.org/10.17577/ijertv6is070249 (date of access: 12.09.2024).

4. Communication Networks S. Retracted: Visual Dynamic Simulation Model of Unstructured Data in Social Networks. *Security and Communication Networks*. 2023. Vol. 2023. P. 1. URL: https://doi.org/10.1155/2023/9759048 (date of access: 12.09.2024).

5. Reframing social media discourse: Converting hate speech to non-hate speech / Y. Kostiuk et al. *Journal of Intelligent & Fuzzy Systems*. 2024. P. 1–14. URL: https://doi.org/10.3233/jifs-219348 (date of access: 12.09.2024).

6. Siegel A. A. Online Hate Speech. *Social Media and Democracy*. 2020. P. 56–88. URL: https://doi.org/10.1017/9781108890960.005 (date of access: 13.09.2024).

7. Hate Speech Prediction on Social Media / I. R. Ammar Aouchiche et al. *SN Computer Science*. 2023. Vol. 4, no. 3. URL: https://doi.org/10.1007/s42979-023-01668-6 (date of access: 16.09.2024).

8. Guiora A., Park E. A. Hate Speech on Social Media. *Philosophia*. 2017. Vol. 45, no. 3. P. 957–971. URL: https://doi.org/10.1007/s11406-017-9858-4 (date of access: 17.09.2024).

9. Zhang Z. Cyberbullying: A Comprehensive Analysis of its Psychological Impact and Preventive Measures. *Journal of Education, Humanities and Social Sciences*. 2024. Vol. 26. P. 655–660. URL: https://doi.org/10.54097/yqsdng51 (date of access: 17.09.2024).

10. Bhatta S. Deepfake, Disinformation and Social Media. *International Journal for Research in Applied Science and Engineering Technology*. 2024. Vol. 12, no. 5. P. 2083–2090. URL: https://doi.org/10.22214/ijraset.2024.61625 (date of access: 19.09.2024).

11. Fake News Detection on Social Media / K. Shu et al. *ACM SIGKDD Explorations Newsletter*. 2017. Vol. 19, no. 1. P. 22–36. URL: https://doi.org/10.1145/3137597.3137600 (date of access: 19.09.2024).

12. Bollen J., Mao H., Pepe A. Modeling Public Mood and Emotion: Twitter Sentiment and Socio-Economic Phenomena. *Proceedings of the International AAAI Conference on Web and Social Media*. 2021. Vol. 5, no. 1. P. 450–453. URL: https://doi.org/10.1609/icwsm.v5i1.14171 (date of access: 20.09.2024).

13. Lemmens J., Markov I., Daelemans W. Improving Hate Speech Type and Target Detection with Hateful Metaphor Features. *Proceedings of the Fourth Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda*, Online. Stroudsburg, PA, USA, 2021. URL: https://doi.org/10.18653/v1/2021.nlp4if-1.2 (date of access: 22.09.2024).

14. Watanabe H., Bouazizi M., Ohtsuki T. Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection. *IEEE Access*. 2023. Vol. 6. P. 13825–13835. URL: https://doi.org/10.1109/access.2018.2806394 (date of access: 24.09.2024).

15. Fortuna P., Nunes S. A Survey on Automatic Detection of Hate Speech in Text. *ACM Computing Surveys*. 2022. Vol. 51, no. 4. P. 1–30. URL: https://doi.org/10.1145/3232676 (date of access: 26.09.2024).

16. Jahan M. S., Oussalah M. A systematic review of Hate Speech automatic detection using Natural Language Processing. *Neurocomputing*. 2023. P. 126232. URL: https://doi.org/10.1016/j.neucom.2023.126232 (date of access: 29.09.2024).

17. Sharma D. K., Singh B., Garg A. An Ensemble Model for detecting Sarcasm on Social Media. *2022 9th International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, 23–25 March 2022.

URL: https://doi.org/10.23919/indiacom54597.2022.9763115 (date of access: 02.10.2024).

18. Lee, S., & Lee, H. (2023). *Language Dynamics in Online Communication: The Use of Slang and Memes*. Social Media Research Journal, 19(4), 201-215 (date of access: 12.10.2024).

19. Zhang, Y., & Chen, Z. (2021). *Cultural Sensitivity in Automated Content Analysis for Social Networks*. International Journal of Language and Culture, 14(2), 99-113 (date of access: 12.10.2024).

20. Kumar, R., & Patel, M. (2020). *Disinformation in the Digital Age: Automated Detection Techniques*. Journal of Digital Media, 7(1), 54-67 (date of access: 15.10.2024).

21. Garcia, M., & Wang, L. (2024). *Balancing Free Speech and Hate Speech on Social Media Platforms*. Media Ethics Review, 29(2), 181-193 (date of access: 16.10.2024).

22. Sarcasm Detection in Tweets: A Feature-based Approach using Supervised Machine Learning Models / A. Rahaman et al. *International Journal of Advanced Computer Science and Applications*. 2021. Vol. 12, no. 6. URL: https://doi.org/10.14569/ijacsa.2021.0120651 (date of access: 16.10.2024).

23. Content Moderation / N. (. Goltz et al. *Real World AI Ethics for Data Scientists*. Boca Raton, 2023. P. 78–90. URL: https://doi.org/10.1201/9781003293125-7 (date of access: 16.10.2024).

24. Zhang, L., Zhao, Y., & Jiang, M. (2020). "The Role of Automated Content Moderation in Social Media Platforms: The Ethical Considerations." *Journal of Information Ethics*, 29(2), 101-119 (date of access: 21.10.2024).

25. Tufekci, Z. (2020). "Algorithmic Harms Beyond Facebook and Google: Emergent Challenges of Computational Agency." *Yale Journal on Regulation*, 32(1), 53-95 (date of access: 22.10.2024).

26. Chen, Y., Zhou, Y., Zhu, S., & Xu, H. (2022). "Detecting offensive language in social media to protect adolescent online safety." *International Conference on Privacy, Security, Risk and Trust*, 71-80 (date of access: 22.10.2024).

27. Developing an online hate classifier for multiple social media platforms / J. Salminen et al. *Human-centric Computing and Information Sciences*. 2020. Vol. 10, no. 1. URL: https://doi.org/10.1186/s13673-019-0205-6 (date of access: 24.10.2024).

28. Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). "Automated hate speech detection and the problem of offensive language." *Proceedings of the Eleventh International AAAI Conference on Web and Social Media*, 512-515 (date of access: 24.10.2024).

29. Dinakar, K., Reichart, R., & Lieberman, H. (2011). "Modeling the detection of textual cyberbullying." *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, 11-17 (date of access: 25.10.2024).

30. LeCun, Y., Bengio, Y., & Hinton, G. (2015). "Deep learning." *Nature*, 521(7553), 436-444. doi:10.1038/nature14539 (date of access: 26.10.2024).

31. Chatzakou, D., Kourtellis, N., Blackburn, J., De Cristofaro, E., Stringhini, G., & Vakali, A. (2017). "Mean birds: Detecting aggression and bullying on Twitter." *Proceedings of the 2017 ACM on Web Science Conference*, 13-22. doi:10.1145/3091478.3091487 (date of access: 26.10.2024).

32. Anderson, H. (2020). "VADER Sentiment Analyzer for Social Media Text." *Computational Tools for Sentiment Analysi* (date of access: 02.11.2024).

33. Badjatiya, P., Gupta, S., Gupta, M., & Varma, V. (2017). "Deep learning for hate speech detection in tweets." *Proceedings of the 26th International Conference on World Wide Web*, 759-760 (date of access: 08.11.2024).

34. Gorrell, G., Greenwood, M. A., Roberts, I., Maynard, D., & Bontcheva, K. (2019). "Twits, twats and twaddle: Trends in online abuse towards UK MPs." *Proceedings of the Tenth ACM Conference on Web Science*, 263-272 (date of access: 08.11.2024).

35. Kwok, I., & Wang, Y. (2013). "Locate the hate: Detecting tweets against blacks." *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, 1621-1622 (date of access: 11.11.2024).

36. Ribeiro, M. H., Calais, P. H., Santos, Y. A., Almeida, V. A., & Meira Jr, W. (2018). "Characterizing and detecting hateful users on Twitter." *Proceedings of the Twelfth International AAAI Conference on Web and Social Media*, 676-679 (date of access: 11.11.2024).

37. Wiegand, M., Ruppenhofer, J., & Kleinbauer, T. (2019). "Detection of abusive language: The problem of biased datasets." *Proceedings of the NAACL-HLT*, 602-608 (date of access: 12.11.2024).

38. Potts, C. (2011). "Sentiment analysis with social media." *Proceedings of the Fifth Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 73-80 (date of access: 14.11.2024).

39. Cambria, E., Schuller, B., Xia, Y., & Havasi, C. (2013). "New avenues in opinion mining and sentiment analysis." *IEEE Intelligent Systems*, 28(2), 15-21. doi:10.1109/MIS.2013.30 (date of access: 24.11.2024).

40. Kshirsagar, M., Walker, M. A., Tree, J. E. F., Anand, P., Abbott, R., & King, J. (2015). "Detecting hidden properties in online discussion with group-level sentiment analysis." *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics*, 56-61 (date of access: 24.11.2024).

41. Pennebaker, J. W., Chung, C. K., Ireland, M., Gonzales, A., & Booth, R. J. (2017). "The development and psychometric properties of LIWC2007." *Austin, TX: LIWC.net* (date of access: 26.11.2024).

42. Characterizing Reddit Submissions That Mention Mental Health (2021). *Proceedings of the International Conference on Web and Social Media (ICWSM)* (date of access: 27.11.2024).

43. Hate Speech in Online Communities: A Linguistic and Network Perspective (2017). *PLOS ONE* (date of access: 27.11.2024).

44. Exploring Public Opinion on Reddit: A Case Study on the COVID-19 Pandemic (2020). *Journal of Medical Internet Research* (date of access: 03.12.2024).

45. Gender Representation in Online Communities: A Study of Reddit (2019). *Proceedings of the ACM on Human-Computer Interaction* (date of access: 04.12.2024).

ДОДАТОК А

Лістинг коду

```
import praw
import pandas as pd
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
from textblob import TextBlob
from transformers import pipeline
import matplotlib.pyplot as plt
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
from tkinter import Tk, Label, Entry, Button, StringVar, OptionMenu, Text, Scrollbar,
      Toplevel, Frame, LEFT, messagebox, END
from matplotlib.animation import FuncAnimation
import sqlite3
from datetime import datetime
import numpy as np
import time

# Default parameters
DEFAULT_CLIENT_ID = '3E3be4CUIVi4O2Dqk4VjWA'
DEFAULT_CLIENT_SECRET = '6VFTR1Y7PO0OmD52lF9ZclaN0fkKxQ'
DEFAULT_USER_AGENT = 'my-reddit-app-v1.0'
DEFAULT_SUBREDDIT = '4chan'
DEFAULT_POST_LIMIT = 3
DEFAULT_MIN_LIKES = 0

# Ініціалізація hate speech detection pipeline
hate_speech_pipeline = pipeline("text-classification", model="unitary/toxic-bert")

# Ініціалізація VADER для аналізу настроїв
vader_analyzer = SentimentIntensityAnalyzer()

# Ініціалізація TextBlob та BERT для аналізу настроїв
sentiment_pipeline   =   pipeline("sentiment-analysis",   model="distilbert-base-uncased-
      finetuned-sst-2-english")

# Глобальні змінні для графіків у реальному часі
sentiment_counts = {'Positive': 0, 'Neutral': 0, 'Negative': 0}
sentiment_values = []
```

```
def init_db():
    """ Initializes the database with a table for toxic comments. """
    conn = sqlite3.connect("toxic_comments.db")
    cursor = conn.cursor()
    cursor.execute('''
        CREATE TABLE IF NOT EXISTS toxic_comments (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            text TEXT,
            author TEXT,
            subreddit TEXT,
            score INTEGER,
            timestamp TEXT,
            toxicity_label TEXT,
            toxicity_score REAL
        )
    ''')
    conn.commit()
    conn.close()

# Функція для збереження токсичних коментарів у базу
def save_toxic_comment(comment, label, score):
    """ Saves the toxic comment data into the database. """
    print(f"Saving comment from subreddit: {comment.get('subreddit', 'Unknown')}")   #
      Debugging line
    conn = sqlite3.connect("toxic_comments.db")
    cursor = conn.cursor()
    cursor.execute('''
        INSERT INTO toxic_comments (text, author, subreddit, score, timestamp,
      toxicity_label, toxicity_score)
        VALUES (?, ?, ?, ?, ?, ?, ?)
    ''', (
        comment['text'],
        comment.get('author', 'Unknown'),
        comment.get('subreddit', 'Unknown'),
        comment.get('score', 0),
        datetime.now().isoformat(),
        label,
        score
```

```
    ))
    conn.commit()
    conn.close()

    print(f"Saved comment from {comment.get('subreddit', 'Unknown')} with toxicity score
      {score}")  # Debugging line

# Функція для збору коментарів з Reddit
def collect_comments(api_id, api_secret, user_agent, subreddit_name, post_limit):
    start_time = time.time()  # Start timing the collection
    comments = []
    try:
        reddit = praw.Reddit(client_id=api_id,
                             client_secret=api_secret,
                             user_agent=user_agent)
        subreddit = reddit.subreddit(subreddit_name)

        for submission in subreddit.hot(limit=post_limit):
            submission.comments.replace_more(limit=0)

            for comment in submission.comments.list():
                comment_data = {
                    'text': comment.body,
                    'score': comment.score,
                    'author': comment.author.name if comment.author else 'Unknown',
                    'subreddit': submission.subreddit.display_name
                }
                comments.append(comment_data)

        end_time = time.time()  # End timing the collection
        collection_time = end_time - start_time
        print(f"Time taken to collect comments: {collection_time:.2f} seconds")

        return comments

    except Exception as e:
        print(f"Error: {e}")
        return []
```

```
# Функція для аналізу настроїв за допомогою VADER
def analyze_sentiment_vader(comments):
    sentiments = []
    for comment in comments:
        analysis = vader_analyzer.polarity_scores(comment['text'])
        if analysis['compound'] >= 0.05:
            sentiments.append('Positive')
            sentiment_counts['Positive'] += 1
        elif analysis['compound'] <= -0.05:
            sentiments.append('Negative')
            sentiment_counts['Negative'] += 1
        else:
            sentiments.append('Neutral')
            sentiment_counts['Neutral'] += 1
        sentiment_values.append(sentiments[-1])
    return sentiments

# Функція для аналізу настроїв за допомогою TextBlob
def analyze_sentiment_textblob(comments):
    sentiments = []
    for comment in comments:
        analysis = TextBlob(comment['text']).sentiment.polarity
        if analysis > 0:
            sentiments.append('Positive')
            sentiment_counts['Positive'] += 1
        elif analysis < 0:
            sentiments.append('Negative')
            sentiment_counts['Negative'] += 1
        else:
            sentiments.append('Neutral')
            sentiment_counts['Neutral'] += 1
        sentiment_values.append(sentiments[-1])
    return sentiments

# Функція для аналізу настроїв за допомогою BERT
def analyze_sentiment_bert(comments):
    sentiments = []
    for comment in comments:
        analysis = sentiment_pipeline(comment['text'])[0]
```

```
        if analysis['label'] == 'POSITIVE':
            sentiments.append('Positive')
            sentiment_counts['Positive'] += 1
        elif analysis['label'] == 'NEGATIVE':
            sentiments.append('Negative')
            sentiment_counts['Negative'] += 1
        else:
            sentiments.append('Neutral')
            sentiment_counts['Neutral'] += 1
        sentiment_values.append(sentiments[-1])
    return sentiments

# Функція для динамічного графіка
def update_graph(i):
    ax.clear()
    ax.bar(sentiment_counts.keys(),  sentiment_counts.values(),  color=['green',  'blue',
      'red'])
    ax.set_title('Розподіл настроїв у реальному часі')
    ax.set_xlabel('Настрої')
    ax.set_ylabel('Кількість')

# Функція для фільтрації за настроєм
def filter_by_sentiment(comments, sentiments, selected_sentiment):
    if selected_sentiment == "All":
        return comments, sentiments
    filtered_comments = []
    filtered_sentiments = []
    for comment, sentiment in zip(comments, sentiments):
        if sentiment == selected_sentiment:
            filtered_comments.append(comment)
            filtered_sentiments.append(sentiment)
    return filtered_comments, filtered_sentiments

# Функція для фільтрації за кількістю лайків
def filter_by_likes(comments, min_likes):
    return [comment for comment in comments if comment['score'] >= min_likes]

# Функція для відображення коментарів у текстовій області
def display_comments(filtered_comments):
```

```
    comment_display.delete(1.0, END)
    for comment in filtered_comments:
        comment_display.insert(END,  f"Лайків:  {comment['score']}\n{comment['text']}\n{'-
      '*50}\n")

# Модифікована функція для аналізу hate speech
def analyze_hate_speech(comments, threshold=0.5):
    hate_speech_results = []
    for comment in comments:
        result = hate_speech_pipeline(comment['text'])[0]
        if "toxic" in result['label'].lower() and result['score'] >= threshold:
            save_toxic_comment(comment, result['label'], result['score'])  # Збереження в
      базу
            hate_speech_results.append((comment, result['label'], result['score']))
    return hate_speech_results

# Функція кнопки для виявлення hate speech
def detect_hate_speech():
    try:
        api_id = api_id_var.get()
        api_secret = api_secret_var.get()
        user_agent = user_agent_var.get()
        subreddit_name = subreddit_var.get()
        post_limit = int(post_limit_var.get())

        if not api_id or not api_secret or not user_agent or not subreddit_name:
            error_message.set("Please fill in all the fields!")
            return

        if post_limit <= 0:
            error_message.set("Post limit must be a positive integer.")
            return

        comments  =  collect_comments(api_id,  api_secret,  user_agent,  subreddit_name,
      post_limit)
        num_comments = len(comments)
        comment_count_label.config(text=f"Зібрано коментарів: {num_comments}")

        if num_comments == 0:
```

```
            error_message.set("No comments collected. Please check the subreddit or API
        credentials.")
            return

        # Аналізує hate speech
        hate_speech_results = analyze_hate_speech(comments)

        # Відобразити результати
        comment_display.delete(1.0, END)
        if hate_speech_results:
            for comment, label, score in hate_speech_results:
                # Check if this exact comment already exists in the database
                conn = sqlite3.connect("toxic_comments.db")
                cursor = conn.cursor()

                # Check for duplicate based on text and other unique identifiers
                cursor.execute('''
                    SELECT COUNT(*) FROM toxic_comments
                    WHERE text = ? AND author = ? AND subreddit = ?
                ''',       (comment['text'],       comment.get('author',       'Unknown'),
        comment.get('subreddit', 'Unknown')))

                duplicate_count = cursor.fetchone()[0]

                if duplicate_count == 0:
                    # Save the comment only if it's not a duplicate
                    save_toxic_comment(comment, label, score)

                comment_display.insert(
                    END, f"Hate Speech Detected!\nScore: {score:.2f}\nLabel: {label}\nText:
        {comment['text']}\n{'-'*50}\n"
                )

                conn.close()
        else:
            comment_display.insert(END, "No hate speech detected.\n")

        error_message.set("")
    except ValueError:
```

```
        error_message.set("Invalid value entered. Please check your inputs.")

def get_toxic_comments():
    conn = sqlite3.connect("toxic_comments.db")
    cursor = conn.cursor()
    cursor.execute('SELECT * FROM toxic_comments')
    comments = cursor.fetchall()
    conn.close()
    return comments

# Оновлена функція для збору коментарів та вибору методу аналізу настроїв
def fetch_comments():
    try:
        api_id = api_id_var.get()
        api_secret = api_secret_var.get()
        user_agent = user_agent_var.get()
        subreddit_name = subreddit_var.get()
        post_limit = int(post_limit_var.get())

        # Перевірка введених значень
        if not api_id or not api_secret or not user_agent or not subreddit_name:
            error_message.set("Please fill in all the fields!")
            return

        if post_limit <= 0:
            error_message.set("Post limit must be a positive integer.")
            return

        # Очищення лічильників перед новим аналізом
        global sentiment_counts
        sentiment_counts = {'Positive': 0, 'Neutral': 0, 'Negative': 0}
        sentiment_values.clear()

        start_processing_time = time.time()  # Start timing the entire processing

        comments = collect_comments(api_id, api_secret, user_agent, subreddit_name,
      post_limit)
        num_comments = len(comments)
        comment_count_label.config(text=f"Зібрано коментарів: {num_comments}")
```

```
  if num_comments == 0:
      error_message.set("No comments collected. Please check the subreddit or API
credentials.")
      return

  # Очищення повідомлення про помилку, якщо збір коментарів успішний
  error_message.set("")

  # Вибір методу аналізу настроїв
  selected_method = analysis_method_var.get()

  sentiment_start_time = time.time()  # Start timing sentiment analysis

  if selected_method == 'VADER':
      sentiments = analyze_sentiment_vader(comments)
  elif selected_method == 'TextBlob':
      sentiments = analyze_sentiment_textblob(comments)
  else:
      sentiments = analyze_sentiment_bert(comments)

  sentiment_end_time = time.time()

  # Фільтрація за настроєм
  selected_sentiment = sentiment_filter.get()
  filtered_comments, filtered_sentiments = filter_by_sentiment(comments, sentiments,
selected_sentiment)

  # Фільтрація за кількістю лайків
  min_likes = int(likes_filter_var.get())
  filtered_comments = filter_by_likes(filtered_comments, min_likes)

  # Виведення коментарів у текстову область
  display_comments(filtered_comments)

  end_processing_time = time.time()  # End timing the entire processing

  # Print out timing information
  print(f"Sentiment Analysis Method: {selected_method}")
```

```
        print(f"Time taken for sentiment analysis: {sentiment_end_time -
      sentiment_start_time:.2f} seconds")
        print(f"Total processing time: {end_processing_time - start_processing_time:.2f}
      seconds")

    except ValueError:
        error_message.set("Invalid value entered. Please check your inputs.")

def fetch_data_for_visualization():
    conn = sqlite3.connect("toxic_comments.db")
    df = pd.read_sql_query("SELECT * FROM toxic_comments", conn)
    conn.close()
    return df

def visualize_data_by_subreddit():
    df = fetch_data_for_visualization()

    # Categorize toxicity scores
    def categorize_toxicity(score):
        if score < 0.3:
            return 'Low Toxicity'
        elif score < 0.7:
            return 'Moderate Toxicity'
        else:
            return 'High Toxicity'

    df['toxicity_category'] = df['toxicity_score'].apply(categorize_toxicity)
    toxicity_categories = df['toxicity_category'].value_counts()

    vis_window = Toplevel(root)
    vis_window.title("Toxicity Categories")
    vis_window.geometry("600x400")

    fig, ax = plt.subplots(figsize=(10, 5))
    toxicity_categories.plot(kind='pie', autopct='%1.1f%%', colors=['green', 'yellow',
      'red'])
    plt.title("Distribution of Toxicity Categories")
    plt.ylabel("")
    plt.tight_layout()
```

```
    canvas = FigureCanvasTkAgg(fig, master=vis_window)
    canvas.get_tk_widget().pack(fill='both', expand=True)
    canvas.draw()

def visualize_toxicity_score_distribution():
    df = fetch_data_for_visualization()

    vis_window = Toplevel(root)
    vis_window.title("Toxicity Score Distribution")
    vis_window.geometry("600x400")

    fig, ax = plt.subplots(figsize=(10, 5))
    df['toxicity_score'].hist(bins=20, edgecolor='black')
    plt.title("Distribution of Toxicity Scores")
    plt.xlabel("Toxicity Score")
    plt.ylabel("Frequency")
    plt.tight_layout()

    canvas = FigureCanvasTkAgg(fig, master=vis_window)
    canvas.get_tk_widget().pack(fill='both', expand=True)
    canvas.draw()

def visualize_toxicity_vs_score_correlation():
    df = fetch_data_for_visualization()

    vis_window = Toplevel(root)
    vis_window.title("Toxicity Score vs. Comment Score")
    vis_window.geometry("800x500")

    fig, ax = plt.subplots(figsize=(12, 6))

    # Scatter plot of toxicity score vs. comment score
    scatter = ax.scatter(df['score'], df['toxicity_score'],
                         alpha=0.7,
                         c=df['toxicity_score'],
                         cmap='YlOrRd')

    # Calculate and plot trend line
```

```
    z = np.polyfit(df['score'], df['toxicity_score'], 1)
    p = np.poly1d(z)
    ax.plot(df['score'], p(df['score']), "r--", label='Trend Line')

    plt.title("Correlation between Comment Score and Toxicity")
    plt.xlabel("Comment Score")
    plt.ylabel("Toxicity Score")
    plt.colorbar(scatter, label='Toxicity Score')
    plt.legend()
    plt.tight_layout()

    canvas = FigureCanvasTkAgg(fig, master=vis_window)
    canvas.get_tk_widget().pack(fill='both', expand=True)
    canvas.draw()

    # Calculate correlation coefficient
    correlation = df['score'].corr(df['toxicity_score'])
    print(f"Correlation coefficient: {correlation}")

# Кнопка для візуалізації графіків
def show_visualization():
    visualize_data_by_subreddit()
    visualize_toxicity_score_distribution()
    visualize_toxicity_vs_score_correlation()


# Функція для запуску графіків у реальному часі
def build_graphs():
    ani.event_source.start()

# Інтерфейс користувача
root = Tk()
root.title('Аналіз настроїв Reddit')
root.geometry("900x700")
root.configure(bg="pink")

api_id_var = StringVar(value=DEFAULT_CLIENT_ID)
api_secret_var = StringVar(value=DEFAULT_CLIENT_SECRET)
user_agent_var = StringVar(value=DEFAULT_USER_AGENT)
```

```
subreddit_var = StringVar(value=DEFAULT_SUBREDDIT)
post_limit_var = StringVar(value=str(DEFAULT_POST_LIMIT))
likes_filter_var = StringVar(value=str(DEFAULT_MIN_LIKES))
error_message = StringVar()
analysis_method_var = StringVar()

Label(root, text="Client ID:", bg="pink").grid(row=0, column=0, padx=10, pady=5)
Entry(root, textvariable=api_id_var, width=50).grid(row=0, column=1, padx=10, pady=5)

Label(root, text="Client Secret:", bg="pink").grid(row=1, column=0, padx=10, pady=5)
Entry(root, textvariable=api_secret_var, width=50).grid(row=1, column=1, padx=10, pady=5)

Label(root, text="User Agent:", bg="pink").grid(row=2, column=0, padx=10, pady=5)
Entry(root, textvariable=user_agent_var, width=50).grid(row=2, column=1, padx=10, pady=5)

Label(root, text="Subreddit:", bg="pink").grid(row=3, column=0, padx=10, pady=5)
Entry(root, textvariable=subreddit_var, width=50).grid(row=3, column=1, padx=10, pady=5)

Label(root, text="Post Limit:", bg="pink").grid(row=4, column=0, padx=10, pady=5)
Entry(root, textvariable=post_limit_var, width=50).grid(row=4, column=1, padx=10, pady=5)

Label(root, text="Likes Filter (Min):", bg="pink").grid(row=5, column=0, padx=10, pady=5)
Entry(root, textvariable=likes_filter_var, width=50).grid(row=5, column=1, padx=10, pady=5)

# Вибір методу аналізу настроїв
Label(root, text="Метод аналізу настроїв:", bg="pink").grid(row=6, column=0, padx=10,
      pady=10)
analysis_method_var.set("VADER")  # Вибір методу за замовчуванням
OptionMenu(root, analysis_method_var, "VADER", "TextBlob", "BERT").grid(row=6, column=1,
      padx=10, pady=10)

# Фільтрація за настроєм
Label(root, text="Фільтр за настроєм:", bg="pink").grid(row=7, column=0, padx=10, pady=10)
sentiment_filter = StringVar()
sentiment_filter.set("All")  # Встановлюємо фільтр за замовчуванням
OptionMenu(root, sentiment_filter, "All", "Positive", "Neutral", "Negative").grid(row=7,
      column=1, padx=10, pady=10)

comment_count_label = Label(root, text="Зібрано коментарів: 0", bg="pink")
```

```
comment_count_label.grid(row=8, column=1, padx=10, pady=5)

Button(root, text="Fetch Comments", command=fetch_comments).grid(row=9, column=1, padx=10,
      pady=1)
Label(root,  textvariable=error_message,  fg="red",  bg="pink").grid(row=10,  column=1,
      padx=10, pady=1)

frame = Frame(root)
frame.grid(row=10, column=0, columnspan=3, pady=5)
# Кнопка hate speech
Button(frame,  text="Detect  Hate  Speech",  command=detect_hate_speech).pack(side=LEFT,
      padx=10)
Button(frame,  text="Візуалізація  даних",  command=show_visualization).pack(side=LEFT,
      padx=10)

# Виведення коментарів
comment_display = Text(root, height=10, width=80)
comment_display.grid(row=11, column=1, padx=10, pady=5)

# Поле для скролінгу тексту
scrollbar = Scrollbar(root)
scrollbar.grid(row=11, column=2, padx=10, pady=5, sticky='ns')
comment_display.config(yscrollcommand=scrollbar.set)
scrollbar.config(command=comment_display.yview)

# Відображення графіків у реальному часі
fig, ax = plt.subplots()
ani = FuncAnimation(fig, update_graph, interval=1000, cache_frame_data=False)
canvas = FigureCanvasTkAgg(fig, root)
canvas.get_tk_widget().grid(row=12, column=1, padx=10, pady=5)

Button(root,  text="Побудувати  графік",  command=build_graphs).grid(row=13,  column=1,
      padx=10, pady=5)

init_db()
root.mainloop()
```