Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Комп'ютерного моделювання та інтелектуальних технологій
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Аналіз тональності коментарів на платформі YouTube з використанням трансформерів
(тема)

Виконав:
здобувач II року навчання,
групи СПРм-24-1
Ткаченко І.Є.
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне проектування
(повна назва освітньої програми)

Керівник: проф. каф. КМІТ Перова І.Г.
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри _____ Гребеннік І. В.
(підпис) (власне ім'я, прізвище)

2026 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Комп'ютерного моделювання та інтелектуальних технологій

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне проектування
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«____» _____ 2026 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Ткаченку Ігорю Євгеновичу
(прізвище, ім'я, по батькові)

1. Тема роботи Аналіз тональності коментарів на платформі YouTube з використанням трансформерів

затверджена наказом по університету від 6 квітня 2026 р. № 268 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 15 травня 2026 р.

3. Вихідні дані до роботи Об'єкт дослідження – процес аналізу текстових даних соціальних мереж (коментарів YouTube) для виявлення емоційного забарвлення. Предмет дослідження – методи автоматичної класифікації тональності текстів на основі сучасних підходів обробки природної мови, зокрема архітектури трансформерів. Вхідні дані – текстові коментарі до відео на YouTube, отримані через офіційний YouTube

4. Перелік питань, що потрібно опрацювати у роботі Вступ. Аналіз предметної області: особливості текстів соціальних мереж (YouTube-коментарів) – неформальність мультимовність, шум, емодзі, сарказм. Огляд існуючих методів аналізу тональності (лексиконні, класичне ML, нейромережі). Обгрунтування вибору трансформерних моделей. Постановка задачі дослідження. Розробка та програмна реалізація: архітектура веб-застосунку (Flask + HTML/CSS/JS). Інтеграція з YouTube Data API v3. Дослідження ефективності: формування тестової вибірки (англомовні та україномовні коментарі), оцінювання якості класифікації, порівняльний аналіз роботи моделі на різних визначення обмежень системи та напрямів подальшого вдосконалення.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних Ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри)_____
Схема методу Bag-of-Words (BoW). Формули розрахунку TF-IDF. Архітектури RNN, LSTM та GRU. Архітектура CNN для обробки текстів. Архітектура трансформера BERT Матриця неточностей для трикласової класифікації. Загальна схема процесу аналізу тональності коментарів YouTube. Схема отримання коментарів через YouTube Data API Етапи очищення та нормалізації тексту коментаря. Поетапний процес токенізації тексту Архітектура XLM-R для аналізу тональності. Класифікація тональності та візуалізація результатів (скріншоти веб-інтерфейсу).

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Розділи спеціальної частини	проф. каф. КМІТ Перова І.Г.		08.05.2026

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / термін використання етапів роботи	Примітка
1	Аналіз предметної області та постановка задачі дослідження	01.04.2026	Виконано
2	Аналіз сучасних методів і підходів до автоматичного визначення тональності тексту	05.04.2026	Виконано
3	Методи та моделі обробки природної мови для визначення тональності текстів	09.04.2026	Виконано
4	Дослідження архітектури трансформерів та їх застосування	14.04.2026	Виконано
5	Розробка та програмна реалізація системи аналізу	18.04.2026	Виконано
6	Дослідження методів попередньої обробки	28.04.2026	Виконано
7	Дослідження ефективності розробленої моделі	02.05.2026	Виконано
8	Перевірка кваліфікаційної роботи на плагіат	05.05.2026	Виконано
9	Попередній захист кваліфікаційної роботи	08.05.2026	Виконано

Дата видачі завдання 01 квітня 2026 р.

Здобувач _____
(підпис)

Керівник роботи _____ проф. каф. КМІТ Ірина ПЕРОВА
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 73 с., 18 Рисунок, 7 табл., 25 джерел.

ВЕБ-ЗАСТОСУНОК, МАШИННЕ НАВЧАННЯ, ОБРОБКА ПРИРОДНОЇ МОВИ, ТОНКЕ НАЛАШТУВАННЯ, ТРАНСФОРМЕРИ, ТОНАЛЬНОСТІ, XLM-R, YOUTUBE, АНАЛІЗ ТОНАЛЬНОСТІ.

Об’єкт дослідження – процес аналізу текстових даних соціальних мереж (коментарів YouTube) для виявлення емоційного забарвлення.

Предмет дослідження – методи автоматичної класифікації тональності текстів на основі сучасних підходів обробки природної мови, зокрема архітектури трансформерів.

Мета роботи – підвищення точності автоматичного визначення тональності багатомовних коментарів YouTube шляхом застосування методу тонкого налаштування (fine-tuning) попередньо навчених трансформерних моделей та реалізація веб-застосунку для демонстрації результатів.

Методи дослідження – системний аналіз предметної області, методи машинного навчання (логістична регресія, LSTM, трансформери), метод тонкого налаштування моделі XLM-R, експериментальне оцінювання з використанням метрик якості класифікації (F1-міра, точність, повнота), а також методи веб-програмування для реалізації клієнт-серверної архітектури.

Сфера застосування – автоматизований моніторинг громадської думки, оцінка реакції аудиторії на відеоконтент, аналітика в медіасфері, системи підтримки прийняття рішень на основі емоційного зворотного зв'язку.

ABSTRACT

An explanatory note to the qualification work of a master: 73 pages, 18 figure, 7 tables, 25 sources.

FINE-TUNING, MACHINE LEARNING, NATURAL LANGUAGE PROCESSING, SENTIMENT ANALYSIS, TRANSFORMERS, WEB APPLICATION, XLM-R, YOUTUBE.

Object of research – the process of analyzing textual data from social media (YouTube comments) to detect emotional coloring.

Subject of research – methods for automatic text sentiment classification based on modern natural language processing approaches, in particular the Transformer architecture.

Aim of the work – improving the accuracy of automatic sentiment detection in multilingual YouTube comments by applying the fine-tuning method to pre-trained Transformer models, and implementing a web application to demonstrate the results.

Research methods – systematic analysis of the subject area, machine learning methods (logistic regression, LSTM, Transformers), the fine-tuning method for the XLM-R model, experimental evaluation using classification quality metrics (F1-score, precision, recall), as well as web programming methods for implementing a client-server architecture.

Application area – automated public opinion monitoring, assessment of audience reaction to video content, media analytics, decision support systems based on emotional feedback.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

API – Application Programming Interface (програмний інтерфейс застосунку).

BERT – Bidirectional Encoder Representations from Transformers (двонаправлений кодувальник на основі трансформерів).

BoW – Bag-of-Words (модель «мішок слів» для представлення тексту).

BPE – Byte-Pair Encoding (метод субслівної токенізації).

CSV – Comma-Separated Values (текстовий формат для зберігання табличних даних).

CNN – Convolutional Neural Network (згорткова нейронна мережа).

CPU – Central Processing Unit (центральний процесор).

Flask – веб-фреймворк для мови Python.

F1 – F1-міра (гармонійне середнє точності та повноти).

GPU – Graphics Processing Unit (графічний процесор).

GRU – Gated Recurrent Unit (керований рекурентний вузол).

HTML – HyperText Markup Language (мова розмітки гіпертексту).

JSON – JavaScript Object Notation (формат обміну даними).

LSTM – Long Short-Term Memory (довга короткочасна пам’ять, різновид рекурентної нейронної мережі).

ML – Machine Learning (машинне навчання).

NLP – Natural Language Processing (обробка природної мови).

RNN – Recurrent Neural Network (рекурентна нейронна мережа).

SVM – Support Vector Machine (метод опорних векторів).

TF-IDF – Term Frequency – Inverse Document Frequency (частота слова – обернена частота документа).

XLM-R – Cross-lingual Language Model – RoBERTa (багатомовна трансформерна модель).

YouTube Data API – офіційний програмний інтерфейс для доступу до даних платформи YouTube.

ВСТУП

У сучасному інформаційному суспільстві соціальні мережі стали невід'ємною частиною комунікації, генеруючи величезні обсяги текстових даних. Платформа YouTube, як один із лідерів серед відеохостингів, акумулює мільйони коментарів, які є цінним джерелом інформації про громадську думку та емоційний стан аудиторії. Ручний аналіз таких масивів даних є практично неможливим через їх обсяг та швидкість появи, що зумовлює необхідність розробки автоматизованих інструментів для визначення емоційного забарвлення тексту, тобто аналізу тональності [1, 2] .

Актуальність теми проекту є надзвичайно високою, оскільки бізнеси, медіа-компанії, блогери та дослідники потребують ефективних інструментів для обробки зворотного зв’язку від аудиторії. Розуміння того, чи є коментарі під відео переважно позитивними, негативними або нейтральними, дозволяє коригувати контент-стратегію, оцінювати репутацію та підвищувати залученість глядачів. Особливістю текстів у соціальних мережах є їхня неформальність, наявність орфографічних помилок, сленгу, емодзі, а також мультимовність (тексти різними мовами та їх змішування). Це створює значні виклики для традиційних методів аналізу, які часто не враховують контекст або орієнтовані на одну мову.

Сучасний розвиток обробки природної мови (Natural Language Processing, NLP) [3, 5] пов'язаний із появою трансформерних моделей, які завдяки механізму самоуваги здатні глибше розуміти контекст. Особливий інтерес становлять багатомовні моделі, такі як XLM-R, що дозволяють ефективно працювати з різномовними даними без створення окремих систем для кожної мови. Отже, використання трансформерів для аналізу тональності є перспективним напрямком, який може забезпечити високу точність навіть на неформальних, зашумлених мультимовних коментарях.

Об'єктом дослідження є процес аналізу текстових даних соціальних мереж. Предметом дослідження є методи автоматичної класифікації

тональності коментарів на основі сучасних підходів обробки природної мови. Основною метою роботи є підвищення точності автоматичного визначення тональності багатомовних коментарів YouTube шляхом застосування тонкого налаштування (fine-tuning) трансформерних моделей та реалізація веб-застосунку для тестування такої системи.

Для досягнення поставленої мети необхідно розв'язати такі завдання:

1. Проаналізувати особливості текстів соціальних мереж та визначити фактори, що впливають на якість аналізу тональності.
2. Виконати огляд існуючих методів аналізу тональності та обґрунтувати вибір трансформерних моделей як основи дослідження.
3. Розробити метод попередньої обробки текстових даних, що враховує специфіку коментарів (шум, мультимовність, емодзі).
4. Запропонувати підхід до класифікації тональності на базі попередньо навченої трансформерної моделі XLM-R з її адаптацією до задачі.
5. Реалізувати веб-застосунок, який інтегрує отримання коментарів через YouTube API, їх обробку моделлю та візуалізацію результатів.
6. Оцінити ефективність запропонованого підходу з використанням метрик якості класифікації та провести порівняльний аналіз на різномовних даних.

Практичне значення роботи полягає у створенні діючого веб-застосунку, який дозволяє автоматизовано аналізувати тональність коментарів до будь-якого публічного відео YouTube. Результати роботи можуть бути використані для моніторингу громадської думки, оцінки репутації та підтримки прийняття рішень у медіасфері, маркетингу та контент-аналітиці.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Теоретичні основи аналізу тональності та його роль у дослідженні текстових даних

Аналіз тональності (sentiment analysis) є одним із напрямів обробки природної мови (Natural Language Processing, NLP) [3, 5], що займається автоматичним виявленням емоційного забарвлення текстових повідомлень [1, 4]. Завдання зводиться до того, щоб на основі змісту тексту визначити, чи висловлює автор позитивне, негативне або нейтральне ставлення до об’єкта мовлення. У більш складних варіантах можливе виділення градацій емоцій (сильний позитив, легке роздратування тощо) або окремих емоцій (радість, гнів, сум). Однак у практичних задачах найчастіше використовують трикласову модель (позитивний, нейтральний, негативний) через її достатність для більшості прикладних сценаріїв.

Особливість аналізу тональності полягає в тому, що текстові дані є неструктурованими: вони не мають заздалегідь заданої форми, містять суб’єктивні оцінки, метафори, сарказм, а також залежать від контексту. Людина легко розпізнає, що фраза «Дуже дякую за цю маячню» має негативний відтінок, тоді як комп’ютерна програма може помилково зарахувати її до позитивних через наявність слова «дякую». Саме в таких випадках потрібне глибше розуміння мови, ніж простий пошук ключових слів.

Роль аналізу тональності в дослідженні текстових даних постійно зростає. Сучасні соціальні мережі, блоги, форуми та відеохостинги генерують величезні обсяги коментарів, які не можна опрацювати вручну. Автоматизований аналіз дозволяє:

- оцінити загальну реакцію аудиторії на певний контент (відео, статтю, новину);
- виявити проблемні місця в продуктах або послугах на основі відгуків;

- відстежувати зміну громадської думки в часі;
- фільтрувати небажані або образливі повідомлення.

Наприклад, власник YouTube-каналу може оперативно дізнатися, який відсоток глядачів відреагував на відео позитивно, а який негативно, без перегляду тисяч коментарів. Дослідники суспільних настроїв можуть аналізувати великі масиви повідомлень щодо політичних подій чи соціальних проблем.

1.2 Особливості текстових даних соціальних мереж та специфіка коментарів платформи YouTube

Текстові дані, що генеруються в соціальних мережах, суттєво відрізняються від традиційних письмових джерел, наукових статей, офіційних документів чи художньої літератури. Якщо останні, як правило, створюються з дотриманням мовних норм, мають чітку структуру та проходять редакційну перевірку, то повідомлення користувачів соцмереж формуються спонтанно, без жорстких правил. Ця свобода висловлювання безпосередньо впливає на ефективність автоматичних методів аналізу.

Платформа YouTube є показовим прикладом такого середовища. Коментарі під відео створюються людьми різного віку, освіти та мовної підготовки, що призводить до високої варіативності текстів. Нижче перелічено ключові характеристики, що ускладнюють аналіз тональності на YouTube.

Користувачі не прагнуть дотримуватися літературних норм. У коментарях часто трапляються скорочення (не «зараз», а «зараз» у неформальному написанні, або «ок» замість «добре»), розмовні та лайливі слова, зневажливі форми. Наприклад, фрази типу «крутяк, бро» або «фу таке собі» несуть чітке емоційне забарвлення, але їх форма відрізняється від словникової.

Наявність орфографічних помилок та спотворен через швидкість набору, неграмотність або навмисне спотворення (інтернет-меми, «олбанська

мова») багато слів пишуться неправильно. «Прекол» замість «прикол», «ватафак» замість «what the fact» – такі варіанти стандартні словники не враховують, тож модель має навчитися їх розпізнавати.

YouTube цее глобальна платформа, під одним відео можуть зустрічатися коментарі українською, англійською, російською, польською, а також їх поєднання в межах одного повідомлення. Явище, коли користувач пише «це було amazing, дуже дякую bro!», називається code-switching. Для автоматичного аналізу це серйозний виклик, оскільки традиційні методи вимагають попереднього визначення єдиної мови тексту.

Смайлики стали невід'ємною частиною інтернет-комунікації. Вони можуть підсилювати емоцію («супер 😍»), змінювати її («дякую 🤬» – явно сарказм) або навіть повністю замінювати текст (ряд із ♡🔥🎉 позначає захват). Ігнорувати емодзі не можна, але й перетворити їх на зрозумілий для моделі вигляд – окреме завдання.

Короткі та контекстуально залежні фрази коментарів складаються з одного-двох слів: «клас», «жах», «😂». Без додаткового контексту (відео, попередні коментарі) зрозуміти тональність іноді неможливо. Наприклад, слово «неймовірно» в коментарі до трейлера фільму-катастрофи може бути як позитивним (якщо фільм вражає), так і негативним (якщо жахливо знято).

Під терміном «Наявність шуму» розуміють елементи, які не несуть смислового навантаження, але заважають аналізу: HTML-теги, посилання, хештеги, повторювані символи («суууууперрр»), зайві пробіли. Їх потрібно видаляти або нормалізувати під час попередньої обробки.

Сарказм та іронія, це мабуть, найскладніша проблема. Автор може використовувати позитивні слова для вираження негативної оцінки: «Дякую, дуже цікаво, аж очі вилазять». Для людини очевидно, що коментар глузливий, але алгоритм, який буквально аналізує лексику, помилиться. Розпізнавання сарказму потребує врахування тону, контексту та знання типових патернів.

Окремо варто зазначити динамічний характер контенту. Нові коментарі з'являються постійно, теми обговорень змінюються, з'являється новий сленг.

Метод аналізу має бути не тільки точним, але й достатньо швидким, щоб обробляти великі масиви даних у прийнятний час, а також здатним адаптуватися до змін мови.

Усі перелічені особливості роблять коментарі YouTube одночасно цінним джерелом думок аудиторії та складним об’єктом для автоматичного аналізу. Врахування цих факторів визначає вибір методів попередньої обробки та архітектури моделі класифікації, які будуть розглянуті в наступних розділах.

Основні труднощі, що виникають при аналізі тональності коментарів на платформі YouTube, можна систематизувати у вигляді узагальненої таблиці. Це дозволяє чітко визначити ключові проблеми та їх вплив на якість роботи моделей обробки природної мови. Відповідні характеристики наведено в таблиці 1.1.

Таблиця 1.1 – Основні проблеми аналізу коментарів YouTube

Проблема	Опис	Вплив на модель
Сленг	Неформальні слова	Помилки класифікації
Емодзі	Неструктурований текст	Складність аналізу
Мультимовність	Різні мови	Потрібна універсальна модель
Сарказм	Прихований сенс	Неправильна інтерпретація

Як видно з таблиці 1.1, найбільший вплив на точність аналізу мають такі фактори, як сарказм, мультимовність та використання неформальної лексики. Це обумовлює необхідність застосування більш складних моделей, здатних враховувати контекст та семантичні особливості тексту.

1.3 Аналіз сучасних методів і підходів до автоматичного визначення тональності тексту

Підходи до аналізу тональності розвивалися поступово, від простих правил до складних нейромережевих архітектур. За час існування цього напряму сформувалося кілька поколінь методів, кожне з яких має свої сильні та слабкі сторони. Для зручності їх можна поділити на три великі групи: лексиконні методи, класичне машинне навчання та підходи на основі нейронних мереж (включаючи трансформери).

Лексиконні методи – найпростіші, вони спираються на заздалегідь складені словники, де кожному слову або виразу приписано числову оцінку тональності (наприклад, +1 для позитивного, -1 для негативного). Аналіз тексту зводиться до підсумовування цих оцінок: якщо сума позитивна – текст позитивний, якщо негативна – негативний, нейтральні слова ігноруються.

Переваги такого підходу очевидні: він не потребує навчання, працює швидко й прозоро – завжди можна побачити, які слова вплинули на результат. Недоліки не менш суттєві: словники не враховують порядок слів, контекст, багатозначність, ігнорують заперечення (фраза «не погано» може бути помилково оцінена як негативна через слово «погано»), а з сарказмом взагалі не справляються.

Класичні методи машинного навчання прийшли на зміну лексиконним. Вони вимагають розмічених даних (набір текстів, де для кожного вже відома правильна тональність). На цих даних алгоритм навчається виявляти статистичні закономірності. Найпоширенішими алгоритмами є наївний байєсівський класифікатор, логістична регресія та метод опорних векторів (SVM).

Перед застосуванням цих методів текст потрібно перетворити на набір числових ознак. Традиційно використовують модель «мішка слів» (Bag-of-Words, BoW) [7] або її вдосконалений варіант TF-IDF [6], який зважує слова за їхньою рідкісністю в корпусі.

Класичні методи дають прийнятну якість на добре підготовлених даних, мають помірні обчислювальні витрати й прості у реалізації. Однак вони не здатні враховувати порядок слів, а їхня ефективність сильно падає на неформальних текстах із помилками та сленгом.

Нейромережеві методи стали наступним кроком: замість фіксованих векторних представлень слів вони використовують щільні ембедінги (word embeddings) [8, 9, 10] – вектори, які навчаються так, що семантично близькі слова опиняються поруч у просторі. Наприклад, вектори слів «добре», «чудово», «супер» будуть близькими, а «погано» – далеко від них.

Для аналізу текстів застосовують різні архітектури. Рекурентні нейронні мережі (RNN) та їхні вдосконалені версії LSTM (Long Short-Term Memory) [11] обробляють текст послідовно, передаючи «пам’ять» від одного слова до наступного. Це дозволяє враховувати попередній контекст. LSTM здатні запам’ятовувати залежності на довгих відрізках тексту, але навчаються повільно й погано масштабуються. Згорткові нейронні мережі (CNN) [12], запозичені з комп'ютерного зору, виділяють локальні патерни (наприклад, типові словосполучення «не дуже», «абсолютно жахливо») незалежно від їх позиції в тексті. Вони швидші за RNN, але не так добре працюють із довгими залежностями.

Трансформерні моделі – найсучасніший підхід, їх ключова відмінність – механізм самоуваги (self-attention), який дозволяє моделі одночасно «бачити» всі слова в реченні та визначати, які з них важливі для розуміння контексту. Наприклад, у реченні «Фільм був нудний, але актори зіграли добре» модель зможе зіставити слово «нудний» із «фільм», а «добре» – з «актори», тоді як попередні архітектури часто плуталися. Трансформери стали основою таких відомих моделей, як BERT, RoBERTa та багатомовної XLM-R.

Ці моделі спочатку навчаються на величезних корпусах текстів (мільйони статей, книг, веб-сторінок), засвоюючи загальні мовні закономірності. Потім їх можна адаптувати до конкретної задачі, наприклад аналізу тональності коментарів YouTube, за допомогою тонкого налаштування

(fine-tuning). Це потребує вже набагато менше розмічених даних, ніж навчання з нуля.

Порівнюючи всі підходи, можна сказати, що лексиконні методи зараз використовуються хіба що для швидкого прототипування. Класичне машинне навчання залишається актуальним, коли обчислювальні ресурси обмежені або потрібна простота інтерпретації. Нейронні мережі (LSTM, CNN) забезпечують вищу точність, але все ще програють трансформерам, особливо на складних, неформальних, мультимовних текстах. Трансформерні моделі на сьогодні є золотим стандартом для аналізу тональності, і саме на них спиратиметься подальше дослідження.

Для більш наочного порівняння розглянутих підходів до аналізу тональності доцільно узагальнити їх основні характеристики у вигляді таблиці. Це дозволяє оцінити сильні та слабкі сторони кожного методу. Порівняльний аналіз наведено в таблиці 1.2.

Таблиця 1.2 – Порівняльний аналіз методів аналізу тональності

Критерій	Лексиконні	ML (SVM)	LSTM/GRU	Transformer
Врахування контексту	Ні	Частково	Так	Так (повністю)
Робота з шумом	Погано	Середньо	Добре	Дуже добре
Мультимовність	Обмежена	Обмежена	Часткова	Висока
Швидкість	Висока	Висока	Середня	Середня
Точність	Низька	Середня	Висока	Найвища

Згідно з таблицею 1.2, традиційні підходи поступаються сучасним нейромережевим моделям за точністю та здатністю працювати зі складними текстовими структурами. Зокрема, трансформерні моделі демонструють найкращі результати завдяки ефективному врахуванню контексту та можливості обробки великих обсягів даних.

Таким чином, розглянуті підходи до аналізу тональності мають як

переваги, так і обмеження. Лексиконні методи характеризуються простотою реалізації та низькими обчислювальними витратами, однак вони не враховують контекст та семантичні зв’язки між словами. Класичні методи машинного навчання (наприклад, SVM або логістична регресія) демонструють кращі результати, проте потребують ручного формування ознак і залишаються чутливими до шуму та неформального стилю тексту.

Нейронні мережі, зокрема рекурентні архітектури (LSTM, GRU), здатні враховувати послідовність слів і контекст, однак мають обмеження при обробці довгих залежностей та потребують значних ресурсів для навчання.

У свою чергу, трансформерні моделі забезпечують більш ефективне врахування контексту за рахунок механізму самоуваги (self-attention), що дозволяє аналізувати взаємозв’язки між усіма словами у реченні незалежно від їх позиції. Це робить їх найбільш придатними для задач аналізу тональності в умовах шумних та мультимовних даних, характерних для платформи YouTube.

1.4 Обґрунтування вибору трансформерних моделей та постановка задачі дослідження

З огляду на проаналізовані в попередніх підрозділах особливості коментарів YouTube та сильні й слабкі сторони різних методів аналізу тональності, необхідно зробити обґрунтований вибір підходу для подальшого дослідження.

Лексиконні методи відпадають через нездатність враховувати контекст, сарказм та багатозначність явища, які в YouTube-коментарях зустрічаються постійно. Класичне машинне навчання (BoW, TF-IDF, логістична регресія, SVM) дає прийнятні результати на академічних корпусах, але показує значно нижчу точність на неформальних, зашумлених текстах із соціальних мереж. Крім того, класичні методи не можуть ефективно працювати з мультимовними даними, оскільки потребують окремого словника для кожної мови.

Рекурентні нейронні мережі (LSTM) та згорткові мережі (CNN) долають проблему врахування послідовності слів, але все ще мають обмеження. LSTM страждають від проблеми зникаючого градієнта на дуже довгих текстах, а CNN захоплюють лише локальні залежності. До того ж, їх навчання потребує великих обсягів розмічених даних, а адаптація до мультимовності вимагає створення окремих моделей або складних архітектур.

Трансформерні моделі [13], зокрема багатомовна XLM-R (XLM-RoBERTa) [16], вирішують більшість зазначених проблем. Їхні попередники – BERT [14] та RoBERTa [15] – довели ефективність архітектури. Їхні переваги для задачі аналізу коментарів YouTube:

1. Глобальне врахування контексту через механізм самоуваги дозволяє правильно інтерпретувати слова в оточенні інших слів, що критично для розуміння сарказму та переносних значень.

2. Мультимовність «з коробки», XLM-R навчалася на текстах 100 мовами, тому вона однаково добре працює з українськими, англійськими, російськими коментарями та їх змішуванням. Не потрібно визначати мову перед аналізом.

3. Стійкість до шуму, авдяки субслівній токенізації (Byte-Pair Encoding) модель може обробляти орфографічні помилки, скорочення та неологізми, розбиваючи їх на відомі підслова.

4. Можливість тонкого налаштування (fine-tuning), замість навчання з нуля, що потребує гігантських обчислювальних ресурсів, модель довчається на відносно невеликому наборі розмічених коментарів, досягаючи високої точності.

5. Відкритість готових моделей, на платформі Hugging Face існують попередньо навчені моделі, вже адаптовані до аналізу тональності. Це значно прискорює розробку.

Саме тому трансформерні моделі обрані як основа для практичної реалізації системи.

Мета дослідження – підвищення точності автоматичного визначення

тональності багатомовних коментарів YouTube шляхом застосування тонкого налаштування трансформерних моделей.

Об’єкт дослідження – процес аналізу текстових даних соціальних мереж.

Предмет дослідження – методи автоматичної класифікації тональності коментарів на основі сучасних підходів обробки природної мови.

Для досягнення поставленої мети необхідно розв’язати такі завдання:

1. Проаналізувати особливості текстів соціальних мереж, зокрема коментарів YouTube, та визначити фактори, що впливають на якість аналізу тональності.

2. Виконати огляд існуючих методів аналізу тональності та обґрунтувати вибір трансформерних моделей.

3. Розробити метод попередньої обробки текстових даних, що враховує специфіку коментарів (шум, мультимовність, емодзі тощо).

4. Запропонувати підхід до класифікації тональності на базі попередньо навченої трансформерної моделі з її адаптацією до задачі.

5. Оцінити ефективність запропонованого підходу з використанням метрик якості класифікації (точність, повнота, F1-міра).

Вхідні дані – текстові коментарі до відео на YouTube, отримані через офіційний YouTube Data API.

Вихідні дані – клас тональності (позитивний, нейтральний, негативний) для кожного коментаря разом із числовою оцінкою впевненості моделі.

З метою обґрунтування вибору трансформерних моделей для вирішення поставленої задачі доцільно узагальнити їх ключові переваги. Це дозволяє чітко продемонструвати їх ефективність у порівнянні з іншими підходами. Основні характеристики наведено в таблиці 1.3.

Таблиця 1.3 – Переваги трансформерних моделей у задачі аналізу тональності

Особливість	Вплив на задачу
Self-attention	Врахування всього контексту

Паралельна обробка	Швидше навчання
Попереднє навчання	Менше даних потрібно
Мультимовність	Робота з різними мовами
Глибоке представлення тексту	Краща точність

Як видно з таблиці 1.3, трансформерні моделі мають ряд суттєвих переваг, зокрема здатність враховувати глобальний контекст, підтримку мультимовності та високу точність класифікації. Це робить їх найбільш доцільним вибором для аналізу тональності коментарів на платформі YouTube.

Отже, на основі проведеного аналізу було обрано трансформерний підхід для реалізації системи аналізу тональності. Основними причинами такого вибору є висока точність класифікації, здатність працювати з мультимовними даними, а також ефективне врахування контексту, що є критично важливим при аналізі коментарів користувачів YouTube.

Крім того, сучасні попередньо навчені моделі (зокрема XLM-R) дозволяють суттєво скоротити час навчання та підвищити якість результатів без необхідності використання великих обсягів розмічених даних.

2 МЕТОДИ ТА МОДЕЛІ ОБРОБКИ ПРИРОДНОЇ МОВИ ДЛЯ ВИЗНАЧЕННЯ ТОНАЛЬНОСТІ ТЕКСТІВ

2.1 Методи попередньої обробки текстових даних з урахуванням їх неструктурованості та шуму

Текстові дані, отримані з соціальних мереж, зокрема коментарі YouTube, надходять у неструктурованому вигляді. Вони містять велику кількість елементів, які не несуть корисного смислового навантаження або навіть заважають подальшому аналізу. Попередня обробка (preprocessing) покликана перетворити «сирий» текст у форму, придатну для подачі в модель машинного навчання, з мінімальною втратою значущої інформації.

Весь процес складається з кількох послідовних кроків, вибір і порядок яких залежить від характеру даних і вибраної моделі. Для коментарів YouTube типовий конвеєр обробки включає такі дії.

Очищення тексту, насамперед видаляються явні «сміттєві» фрагменти: HTML-теги (якщо коментарі зберігалися з розміткою), URL-посилання (вони рідко несуть емоційне навантаження), зайві пробіли, спеціальні символи типу & або \n. Деякі символи пунктуації можуть бути видалені, але не всі, наприклад, знаки оклику чи питання часто підсилюють емоцію, тому їх краще зберігати або замінювати на спеціальні токени.

Нормалізація регістру, більшість моделей (особливо трансформери) не чутливі до регістру після того, як їх навчили на текстах у різних варіантах. Однак для зменшення розміру словника та уникнення дублювання («Добре», «добре», «ДОБРЕ») зазвичай приводять усі символи до нижнього регістру. Виняток можуть становити акроніми або слова, де регістр несе смислове навантаження (наприклад, «УКРАЇНА» у крикливому коментарі), але на практиці для YouTube-коментарів цей крок безпечний.

Обробка емодзі які являються потужнім джерелом емоційної інформації, тому їх не можна просто відкидати. Використовують два основних

підходи: заміна емодзі на текстовий опис (😊 → усмішка, ♡ → серце) або збереження їх як окремих токенів. Для многомовної моделі XLM-R другий варіант є прийнятним, оскільки токенізатор моделі здатен обробляти Unicode-символи. У лістингу 2.1 показано приклад заміни емодзі на текстові дескриптори.

Лістинг 2.1 – Функція заміни емодзі на текстові описи

```
import emoji
def replace_emojis(text: str) -> str:
    return emoji.demojize(text, delimiters=(" ", " "))
```

Видалення стоп-слів які являються поширеною службою частини мови (сполучники, прийменники, частки), які рідко впливають на тональність. У класичному машинному навчанні їх вилучали для зменшення кількості ознак. Для трансформерів цей етап не є обов’язковим, оскільки механізм уваги сам навчається ігнорувати неважливі слова. Проте в деяких випадках видалення стоп-слів може знизити шум у дуже коротких коментарях.

Лематизація та стемінг – це зведення різних форм слова до однієї основи. Стемінг – це грубіший метод, який просто відрізає закінчення (наприклад, «граю», «граєш», «грав» → «гр»). Лематизація використовує словник і граматичні правила для отримання нормальної форми («граю», «граєш», «грав» → «грати»). Для сучасних моделей, які вже навчені на великих корпусах, цей крок зазвичай пропускають, оскільки модель здатна узагальнити різні форми самостійно. Однак для класичних методів (логістична регресія, SVM) лематизація допомагає зменшити розмір словника й підвищити точність.

Токенізація – це розбиття тексту на окремі елементи, токени. У найпростішому випадку токенами є слова, розділені пробілами. Сучасні моделі використовують субслівну токенізацію (WordPiece, BPE), яка розбиває рідкісні слова на поширені підслова. Наприклад, слово багатомовний може бути розбите на багато, мов, ний. Це дозволяє обробляти слова, яких модель

не бачила під час навчання, а також справлятися з орфографічними помилками.

Усі описані кроки застосовуються в різних комбінаціях залежно від вибраної моделі. Для трансформерних моделей, як правило, обмежуються очищенням, нормалізацією регістру, обробкою емодзі та токенізацією (яку виконує спеціалізований токенізатор). Стемінг, лематизацію та видалення стоп-слів залишають для класичних методів, оскільки для трансформерів вони можуть навіть погіршити результат через втрату контексту.

2.2 Підходи до представлення тексту у вигляді числових ознак для задач машинного навчання

Будь-який алгоритм машинного навчання працює з числами, а не з текстом. Тому перед тим, як подати коментар на вхід моделі, його потрібно перетворити на вектор числових ознак. Від того, наскільки вдало це перетворення передає смислові та емоційні характеристики тексту, прямо залежить якість класифікації. За роки розвитку обробки природної мови склалося кілька основних підходів до такого представлення.

Модель «мішка слів» (Bag-of-Words, BoW) – найпростіший і найстаріший спосіб. Будується словник усіх унікальних слів, що зустрічаються в навчальному корпусі. Кожен текст подається вектором, довжина якого дорівнює розміру словника, а значення кожної координати, кількість входжень відповідного слова в текст. Порядок слів повністю втрачається, тобто фрази «гарний фільм» і «фільм гарний» отримають однаковий вектор. Незважаючи на грубість такого підходу, BoW досі використовується як базовий метод через простоту та швидкість. Схема методу Bag-of-Words наведена на рисунку 2.1.

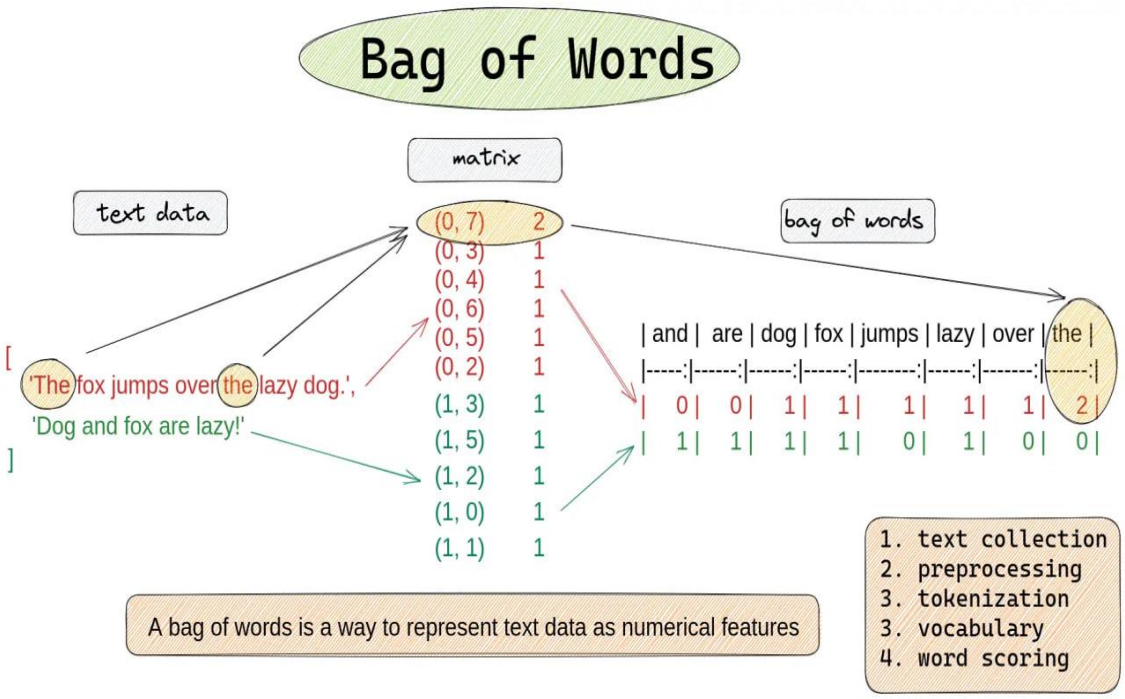

Рисунок 2.1 - Схема методу Bag of Words (BoW)

TF-IDF (Term Frequency – Inverse Document Frequency) – удосконалення BoW, яке враховує не лише частоту слова у конкретному тексті, але й його рідкісність у всьому корпусі. Ідея в тому, що слова, які трапляються майже в кожному документі (наприклад, «і», «але», «цей»), малоінформативні, тоді як рідкісні слова («чудово», «жахливо») краще розрізняють класи. Формула TF-IDF складається з двох множників:

1. TF – відносна частота слова в документі (скільки разів слово зустрілося, поділене на загальну кількість слів);

2. IDF – логарифм відношення загальної кількості документів до кількості документів, що містять це слово.

Схема методу Term Frequency-Inverse Document Frequency наведена на рисунку 2.2.

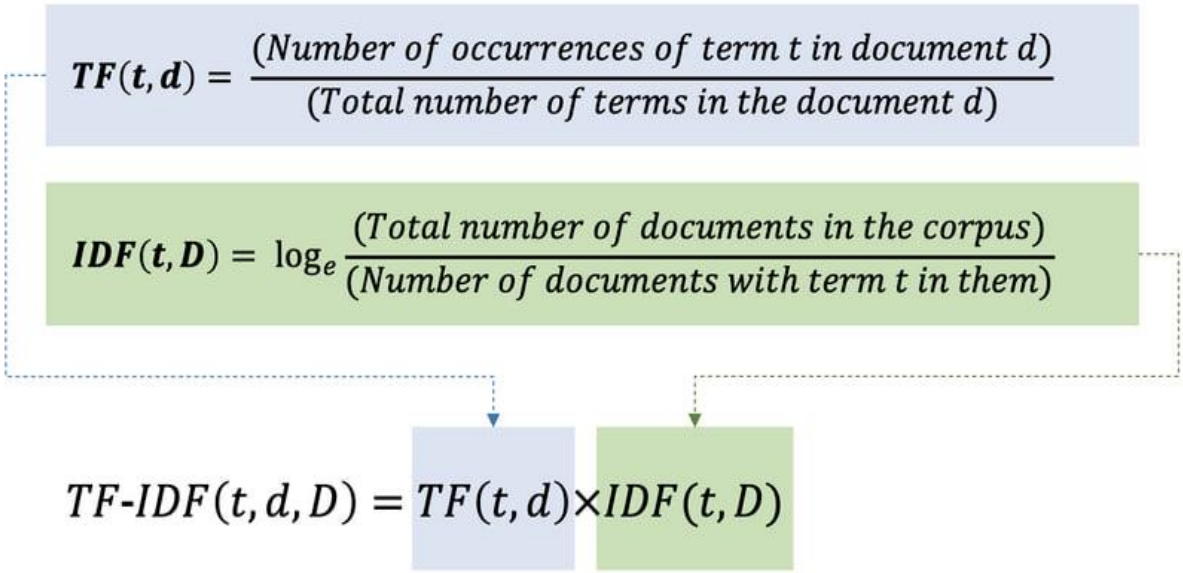

Рис 2.2 - Формули розрахунку TF-IDF

І BoW, і TF-IDF генерують дуже розріджені вектори (більшість координат, нулі), оскільки в одному тексті використовується лише невелика частка слів зі словника. Розмірність такого простору може сягати десятків або сотень тисяч ознак, що створює проблеми для деяких алгоритмів.

Векторні представлення слів (Word Embeddings) – на противагу розрідженим векторам, ембедінги є щільними векторами фіксованої невеликої розмірності (зазвичай 100–300). Кожне слово відображається в точку в цьому просторі так, що семантично близькі слова опиняються поруч. Наприклад, вектори слів «кіт», «пес», «хом'як» будуть згруповані разом, а вектор «автомобіль» далеко від них. Ембедінги навчаються на великих текстових корпусах без розмітки (неконтрольоване навчання) за допомогою таких методів, як Word2Vec або GloVe.

Для отримання вектора цілого тексту з ембедінгів слів застосовують агрегацію: додають або усереднюють вектори всіх слів у тексті. Цей підхід уже враховує семантику, але все ще втрачає порядок слів.

Контекстні ембедінги (трансформери) – сучасні моделі, зокрема BERT та XLM-R, не використовують окремий етап перетворення тексту на вектор. Замість цього вони мають вбудований шар ембедінгів, який навчається разом

з усією мережею. Ці ембедінги є контекстними: вектор слова «банк» у реченнях «банк даних» і «фінансовий банк» буде різним, оскільки модель враховує оточення. Для класифікації тексту зазвичай використовують вектор спеціального токена [CLS], який агрегує інформацію про весь вхід.

У лістингу 2.2 показано, як за допомогою бібліотеки transformers отримати векторне представлення тексту для моделі XLM-R (без шару класифікації).

Лістинг 2.2 – Отримання контекстного векторного представлення тексту за допомогою XLM-R

```
from transformers import AutoTokenizer, AutoModel
tokenizer = AutoTokenizer.from_pretrained("xlm-roberta-base")
model = AutoModel.from_pretrained("xlm-roberta-base")
inputs = tokenizer("Цей фільм чудовий", return_tensors="pt")
outputs = model(**inputs)
text_embedding = outputs.last_hidden_state[:, 0, :]  # токен
[CLS]
```

Вибір підходу залежить від задачі та доступних ресурсів. Для класичних методів (логістична регресія, SVM) достатньо TF-IDF. Для рекурентних нейромереж використовують статичні ембедінги Word2Vec або GloVe. Для трансформерів не потрібно окремо турбуватися про представлення тексту, модель навчається будувати його самостійно під час тонкого налаштування. У цьому дослідженні застосовано саме трансформерний підхід, оскільки він дає найкращі результати на мультимовних, неформальних текстах.

2.3 Класичні та нейромережеві методи класифікації текстових даних

Після того як текст перетворено на набір числових ознак, наступним кроком є власне класифікація, віднесення вектора до одного з класів (позитивний, нейтральний, негативний). Існує широкий спектр алгоритмів, які відрізняються складністю, швидкістю та здатністю виявляти приховані закономірності. У цьому підрозділі розглянуто класичні методи машинного

навчання та ранні нейромережеві архітектури (трансформери, як найсучасніший підхід, винесено в окремий підрозділ 2.4).

Класичні методи машинного навчання – до них відносять алгоритми, які працюють із векторами ознак фіксованої розмірності, отриманими, наприклад, через BoW або TF-IDF. Вони не враховують порядок слів, але мають низьку обчислювальну вартість і добре інтерпретуються.

Наївний байєсівський класифікатор (Naive Bayes) ґрунтується на теоремі Байєса та припущенні про незалежність ознак (слів) між собою. Попри наївність цього припущення, алгоритм показує хороші результати в текстових задачах, особливо на великих корпусах. Його переваги – це швидкість навчання та передбачення, простота реалізації.

Логістична регресія (Logistic Regression) – це лінійний класифікатор, який оцінює ймовірність належності до кожного класу. Для багатокласової задачі використовується розширення softmax. Логістична регресія часто дає кращі результати, ніж наївний байєсівський класифікатор, але потребує більше даних для стабільного навчання.

Метод опорних векторів (Support Vector Machine, SVM) будує гіперплощину, яка максимально розділяє класи у просторі ознак. Завдяки використанню ядерних функцій (kernel trick) SVM здатен працювати з нелінійними розділеннями. Він добре справляється з високовимірними текстовими даними й часто слугує надійним базовим рівнем.

У лістингу 2.3 показано приклад навчання логістичної регресії на TF-IDF ознаках за допомогою бібліотеки scikit-learn [24].

Лістинг 2.3 – Навчання логістичної регресії на TF-IDF ознаках

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
vectorizer = TfidfVectorizer(max_features=10000)
X_train = vectorizer.fit_transform(texts_train)
model = LogisticRegression(max_iter=1000)
model.fit(X_train, labels_train)
```

Нейромережеві методи (до трансформерів) – з появою глибокого

навчання класичні алгоритми почали поступатися місцем нейронним мережам, які здатні автоматично виявляти складні, нелінійні залежності та враховувати структуру тексту.

Рекурентні нейронні мережі (RNN) спеціально розроблені для роботи з послідовностями. Вони обробляють текст слово за словом, передаючи прихований стан, який несе інформацію про прочитаний контекст. Однак базові RNN страждають від проблеми зникаючого градієнта: чим довша послідовність, тим менший вплив перших слів на кінцевий результат.

LSTM (Long Short-Term Memory) та GRU (Gated Recurrent Unit) – модифікації RNN, які вводять спеціальні вентильні механізми. Вони дозволяють вибірково «забувати» непотрібну інформацію та «запам’ятовувати» важливу на довгих відстанях. LSTM успішно застосовуються для аналізу тональності, оскільки можуть враховувати вплив слів, розділених десятками позицій. Схема методу Term Frequency-Inverse Document Frequency наведена на рисунку 2.2.

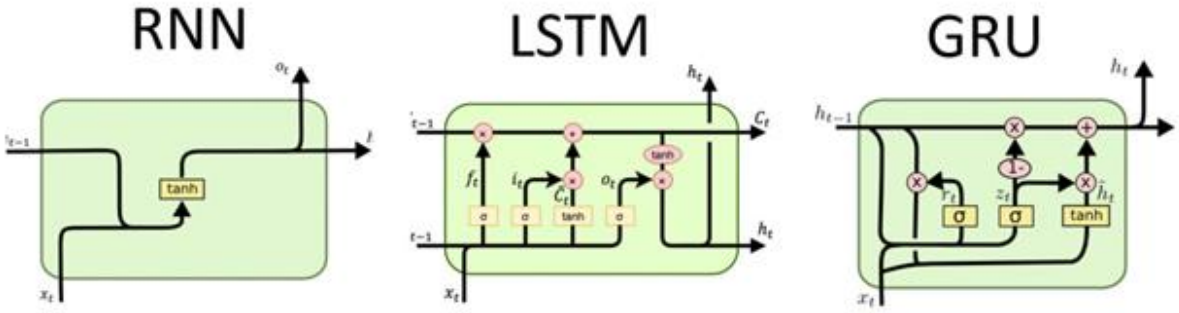

Рисунок 2.3 - Архітектури RNN, LSTM та GRU

Згорткові нейронні мережі (CNN), запозичені з комп’ютерного зору, обробляють текст за допомогою одновимірних згорток. Фільтри (ядра) ковзають по послідовності ембедінгів слів, виявляючи локальні патерни, характерні n-грами (наприклад, «не дуже», «дуже добре», «жахливо погано»). Потім шар пулінгу агрегує найсильніші ознаки. CNN навчаються швидше за RNN і добре виділяють типові словосполучення, але не враховують

довготривалі залежності. Схема мережі CNN наведена на рисунку 2.4.

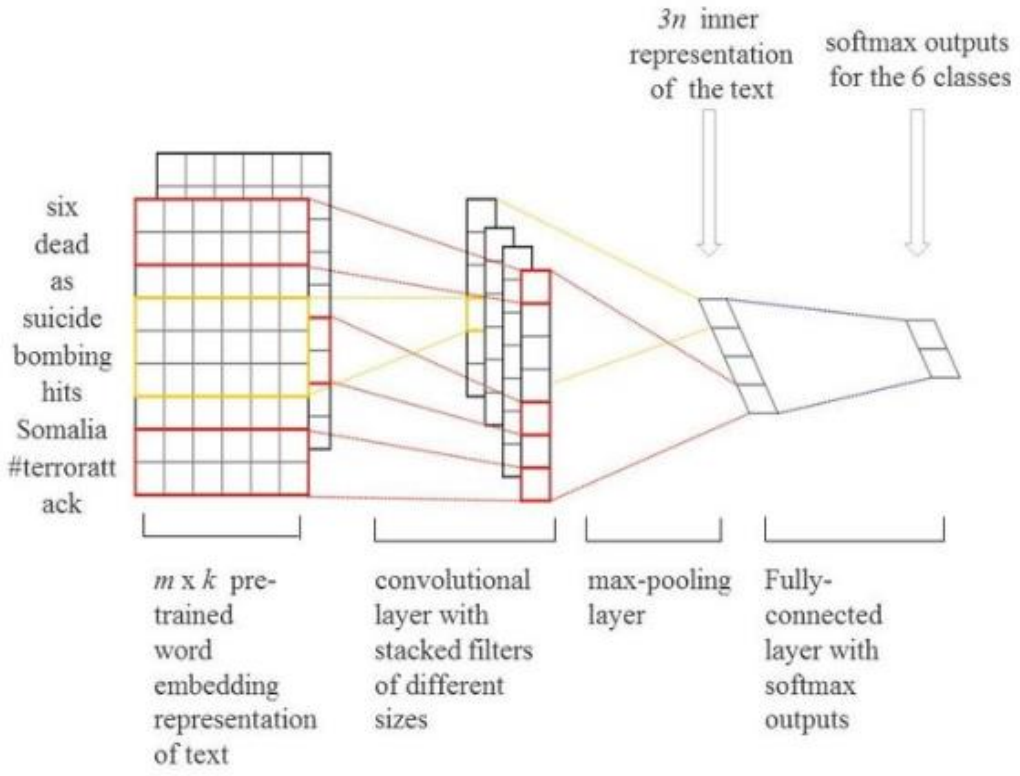

Рисунок 2.4 - Архітектура CNN для обробки текстів

Незважаючи на свої переваги, описані підходи мають суттєві недоліки для аналізу коментарів YouTube. Класичні методи втрачають порядок слів і не враховують контекст. RNN (навіть LSTM) обробляють текст послідовно, що обмежує паралелізацію та ускладнює врахування довготривалих залежностей, коли важливі слова розташовані далеко. CNN добре бачать локальні патерни, але «не розуміють» структуру всього речення. Крім того, більшість із цих методів одномовні й потребують окремої адаптації для кожної мови, що неприйнятно для багатомовних YouTube-коментарів. Саме ці обмеження привели до появи трансформерів, які долають більшість зазначених проблем.

2.4 Архітектура трансформерів та їх застосування у задачах аналізу тональності

Попередні нейромережеві архітектури такі як RNN, LSTM, CNN – мали принципові обмеження: рекурентні мережі обробляли текст послідовно, що не дозволяло їм «бачити» всі слова одночасно, а згорткові мережі захоплювали лише локальні залежності. Трансформери, запропоновані у 2017 році в статті «Attention Is All You Need» [13], запровадили зовсім інший підхід, який швидко став домінуючим у всіх задачах обробки природної мови. Для глибшого розуміння механізму самоуваги корисним є ілюстрований посібник [17].

Механізм самоуваги (self-attention) – центральний елемент трансформера – це механізм уваги, який дозволяє моделі визначати важливість кожного слова відносно всіх інших слів у реченні. Для кожного слова обчислюються три вектори: запиту (query), ключа (key) та значення (value). Вага уваги між двома словами обчислюється як скалярний добуток вектора запиту одного слова з вектором ключа іншого. Чим більший добуток, тим сильніше одне слово «уважає» на інше. Потім ці ваги нормалізуються через softmax, і підсумкове представлення слова отримується як зважена сума векторів значень усіх слів.

Багатоголова увага (multi-head attention) – замість одного механізму уваги трансформер використовує кілька, що працюють паралельно. Кожна «голова» вивчає різні типи залежностей: одна може фіксувати граматичні зв’язки, інша – семантичні, третя – позиційні. Результати всіх голів об’єднуються та лінійно перетворюються.

Позиційне кодування – оскільки механізм уваги не враховує порядок слів (він бачить усі слова одночасно), у трансформер додають позиційні кодування, спеціальні вектори, які несуть інформацію про місце слова в реченні. Вони додаються до ембедінгів слів перед подачею в перший шар.

Трансформер складається з послідовних однакових блоків, кожен із яких

містить шар багатоголової уваги, позиційний прямозв'язний шар (feed-forward) та залишкові з'єднання з нормалізацією. Для задач класифікації використовується лише енкодерна частина (як у BERT), а для генерації тексту повна архітектура енкодер-декодер.

BERT (Bidirectional Encoder Representations from Transformers) та її варіанти: стала першою моделлю, яка продемонструвала перевагу двонаправленого контексту. Вона навчається на двох завданнях: маскування слів (MLM), передбачити замасковане слово за контекстом, та передбачення наступного речення (NSP), визначити, чи є одне речення продовженням іншого. RoBERTa – це вдосконалена версія BERT, яка відмовилася від завдання NSP, збільшила обсяг навчальних даних і застосувала динамічне маскування. Як наслідок, RoBERTa стабільно перевершує BERT на більшості тестових наборів. Архітектура трансформера BERT наведена на рисунку 2.5.

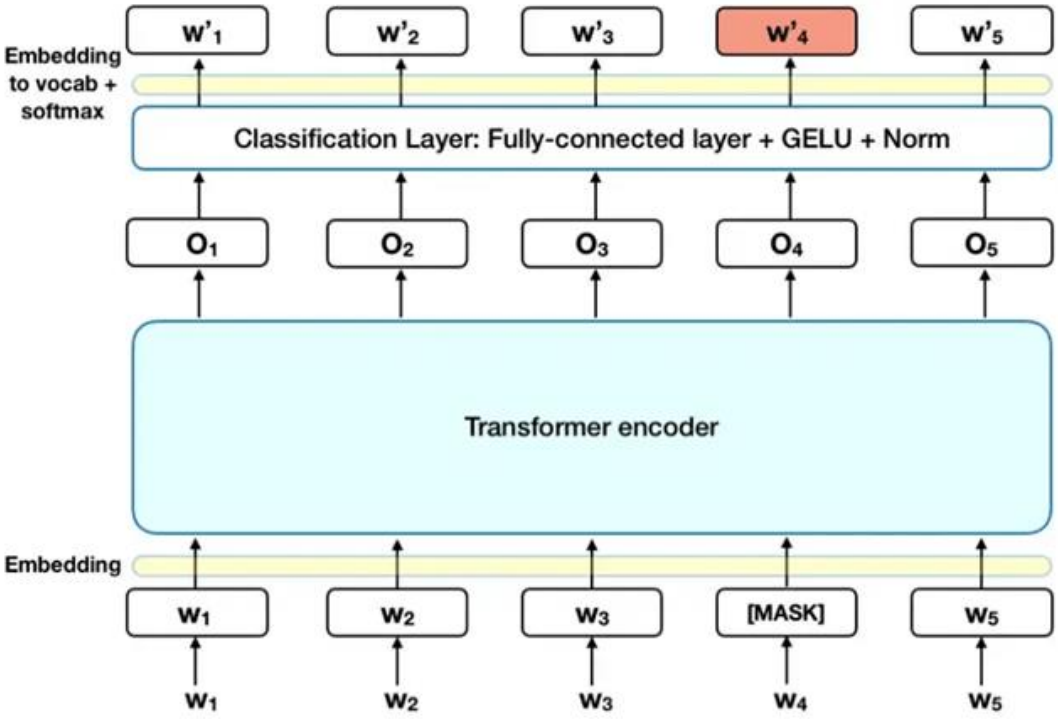

Рисунок 2.5 - Архітектура трансформерів (BERT)

XLM-R (Cross-lingual Language Model – RoBERTa) [16] – для цього дослідження ключове значення має саме XLM-R. Для роботи з моделлю

використовується бібліотека transformers від Hugging Face [18, 19]. Це багатомовна модель, навчена на текстах 100 мовами (включаючи українську, англійську, російську, польську). Її архітектура ідентична RoBERTa, але навчальний корпус значно ширший. Завдяки цьому XLM-R здатна без додаткового налаштування працювати з різномовними текстами та навіть зі змішуванням мов у межах одного речення.

Щоб використати трансформерну модель для класифікації тональності, до її архітектури додають класифікаційну «голову», один або кілька повнозв'язних шарів, які перетворюють вихідний вектор (зазвичай той, що відповідає спеціальному токену [CLS]) на вектор ймовірностей класів. Далі модель піддається тонкому налаштуванню (fine-tuning) на розмічених даних: усі її параметри (і базової архітектури, і класифікаційної голови) оновлюються під час навчання. Цей процес потребує значно менше обчислювальних ресурсів і розмічених даних, ніж навчання з нуля.

Для цього дослідження обрана не базова XLM-R, а її версія, вже донавчена на YouTube-коментарях (AmaanP314/youtube-xlm-roberta-base-sentiment-multilingual). Це означає, що модель уже бачила тисячі прикладів коментарів різними мовами та може бути використана без додаткового навчання для отримання прийнятної якості, або ж донавчена на власних даних для покращення результатів.

Таким чином, трансформерні моделі, особливо багатомовна XLM-R, є оптимальним вибором для аналізу тональності коментарів YouTube завдяки здатності враховувати глобальний контекст, працювати з мультимовними даними та ефективно адаптуватися до конкретного завдання через тонке налаштування.

2.5 Метрики оцінювання якості моделей класифікації

Після того як модель навчена (або налаштована), необхідно перевірити, наскільки добре вона виконує поставлене завдання. Для задач класифікації,

зокрема аналізу тональності, існує набір стандартних метрик, які дозволяють оцінити різні аспекти якості: загальну точність, здатність знаходити об’єкти певного класу, уникнення хибних спрацьовувань. Усі метрики базуються на порівнянні передбачень моделі з істинними мітками на тестовій вибірці.

Матриця неточностей (confusion matrix) – це таблиця, рядки якої відповідають реальним класам, а стовпці, передбаченим. Для трикласової задачі (позитивний, нейтральний, негативний) вона має розмір 3×3. Діагональні елементи показують кількість правильно класифікованих прикладів для кожного класу, а позадіагональні, помилки (наприклад, позитивний коментар, помилково віднесений до нейтральних). На рисунку 2.6 показано приклад матриці неточностей для гіпотетичної моделі аналізу тональності.

Фактичний клас \ Прогнозований клас	**Позитивний**	**Нейтральний**	**Негативний**
Позитивний	TP_pos	FN_pos_neu	FN_pos_neg
Нейтральний	FP_neu_pos	TP_neu	FN_neu_neg
Негативний	FP_neg_pos	FP_neg_neu	TP_neg

Рисунок 2.6 – Матриця неточностей для трикласової класифікації тональності

Для оцінювання якості моделі класифікації було використано матрицю неточностей (confusion matrix), яка відображає кількість правильних та помилкових класифікацій для кожного класу. Матриця наведена в таблиці 2.1.

З отриманих результатів видно, що модель найкраще класифікує позитивні та негативні коментарі, тоді як найбільша кількість помилок

припадає на нейтральний клас. Це пояснюється тим, що нейтральні повідомлення часто мають слабко виражене емоційне забарвлення, що ускладнює їх точну класифікацію.

Таблиця 2.1 – Матриця неточностей для трикласової класифікації тональності

Фактичний клас \ Прогнозований клас	Позитивний	Нейтральний	Негативний
Позитивний	120	15	5
Нейтральний	10	95	20
Негативний	8	12	110

Найпростішою метрикою є точність класифікації (accuracy) – частка правильно класифікованих прикладів від загальної кількості:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Ця метрика є зрозумілою та зручною, але має обмеження: у випадку нерівномірного розподілу класів (наприклад, коли позитивних коментарів значно більше, ніж негативних) модель може демонструвати високу точність, водночас погано визначаючи менш представлені класи.

Для більш детального аналізу використовуються метрики precision та recall.

Precision (точність) [20] відображає, яка частка об'єктів, віднесених моделлю до певного класу, дійсно належить до цього класу. Іншими словами, це міра «чистоти» прогнозів моделі. Для класу positive precision обчислюється як:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall (повнота) [20] показує, яку частину реальних об'єктів певного класу модель змогла правильно виявити. Матриця неточностей (confusion matrix) [21] дає змогу детально проаналізувати помилки моделі. Це міра «чутливості» моделі до об’єктів цього класу:

$$\text{Recall} = \frac{TP}{TP + FP}$$

Між precision та recall часто існує компроміс: підвищення одного показника може призводити до зниження іншого. Для їх узагальнення використовується *F1-міра* (F1-score), яка є гармонійним середнім цих двох показників:

$$\text{F1} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

F1-міра дозволяє отримати збалансовану оцінку якості моделі, враховуючи як точність, так і повноту. У задачах аналізу тональності вона часто використовується як основний критерій якості.

У випадку трикласової класифікації (позитивний, нейтральний, негативний) метрики можуть обчислюватися окремо для кожного класу, а потім усереднюватися. Існують різні способи усереднення:

1. Macro-average (макроусереднення) – обчислюється середнє арифметичне значення метрики за всіма класами. Кожен клас має однакову вагу незалежно від кількості об’єктів у ньому.

2. Weighted-average (зважене усереднення) – обчислюється середнє з урахуванням кількості об’єктів у кожному класі. Це дозволяє отримати більш об’єктивну оцінку, особливо при нерівномірному розподілі даних.

Вибір метрик залежить від особливостей задачі та цілей дослідження. Для аналізу коментарів YouTube важливо не лише загальна точність, але й здатність моделі правильно визначати різні типи емоційних реакцій. Зокрема:

- нейтральні коментарі часто є найбільш чисельними, але їхнє правильне визначення не є критичним порівняно з помилковою класифікацією позитивних або негативних повідомлень;

- вартість помилки може бути різною залежно від класу (наприклад, негативний коментар, помилково віднесений до позитивних, може спотворити оцінку сприйняття контенту).

З огляду на це, доцільно використовувати кілька метрик одночасно: accuracy для загальної оцінки, а також precision, recall та F1-score для кожного класу окремо. Основним критерієм якості в даному дослідженні обрано макроусереднену F1-міру, яка забезпечує збалансовану оцінку роботи моделі на всіх класах.

У лістингу 2.4 показано, як обчислити ці метрики за допомогою бібліотеки scikit-learn.

Лістинг 2.4 – Обчислення метрик класифікації за допомогою scikit-learn

```
from sklearn.metrics import classification_report
y_true = ["positive", "neutral", "negative", "positive"]
y_pred = ["positive", "neutral", "negative", "neutral"]
print(classification_report(y_true, y_pred))
```

Для оцінювання якості моделі на коментарях YouTube доцільно використовувати кілька метрик одночасно. Accuracy дасть загальне уявлення про роботу системи. Precision, recall та F1 для кожного класу дозволять побачити, чи модель однаково добре справляється з усіма типами тональності (часто нейтральні коментарі визначаються гірше, ніж полярні). Основним критерієм порівняння різних підходів у цій роботі обрано macro F1-score, оскільки він не залежить від дисбалансу класів і дає чесну оцінку здатності моделі до узагальнення.

2.6 Функціональні та нефункціональні вимоги до системи аналізу тональності

Перш ніж приступити до реалізації системи, необхідно чітко визначити, що саме вона має робити (функціональні вимоги) і за якими якісними характеристиками це має відбуватися (нефункціональні вимоги). Такий підхід дозволяє уникнути неоднозначностей під час розробки та забезпечує відповідність кінцевого продукту поставленим задачам.

Функціональні вимоги описують конкретні дії, які система повинна виконувати.

1. Отримання коментарів за посиланням на відео – користувач вводить URL відео YouTube (підтримуються формати `youtu.be/*`, `youtube.com/watch?v=*`, `youtube.com/shorts/*`). Система за допомогою YouTube Data API отримує тексти коментарів до цього відео. Користувач має змогу обмежити кількість коментарів (наприклад, 50, 100, 200, 500).
2. Класифікація тональності – отримані коментарі автоматично обробляються трансформерною моделлю (на основі XLM-R, попередньо навченою на YouTube-коментарях). Для кожного коментаря система визначає один із трьох класів: «позитивний», «нейтральний» або «негативний», а також обчислює числову оцінку впевненості (від 0 до 1).
3. Відображення статистики – після аналізу система показує узагальнені підсумки: кількість і відсоток позитивних, нейтральних, негативних коментарів, а також загальну кількість опрацьованих повідомлень. Статистика візуалізується у вигляді кругової діаграми.
4. Перегляд списку коментарів – користувач бачить таблицю з текстом кожного коментаря, визначеним класом тональності (з кольоровим маркуванням) та рівнем впевненості моделі (у відсотках, із графічною шкалою).

5. Фільтрація – передбачено можливість відфільтрувати таблицю, залишивши лише коментарі певного класу (позитивні, нейтральні або негативні). Фільтр застосовується миттєво без повторного звернення до сервера.

6. Пагінація – оскільки коментарів може бути кілька сотень, список розбивається на сторінки (наприклад, по 8–10 коментарів). Користувач може гортати сторінки вперед/назад.

7. Експорт результатів – система надає можливість зберегти результати аналізу у файл. Підтримуються два формати: CSV (для відкриття в табличних редакторах) та JSON (для подальшої програмної обробки). Файл включає текст коментаря, клас тональності та значення впевненості.

Нефункціональні вимоги визначають обмеження та критерії якості, яким має відповідати система.

1. Продуктивність і час відгуку – система повинна обробляти пакет зі 200 коментарів не довше ніж за 30–40 секунд на типовому апаратному забезпеченні (з використанням GPU або CPU із достатньою пам’яттю). Інтерактивні дії (фільтрація, пагінація, відкриття сторінки) мають виконуватися миттєво, без затримок.

2. Надійність – система коректно обробляє помилки доступу до YouTube API (невірне посилання, закрите відео, перевищення квоти, відсутність коментарів). У разі помилки користувач отримує зрозуміле повідомлення без «падіння» всього застосунку.

3. Масштабованість – хоча система орієнтована на аналіз окремих відео, її архітектура дозволяє збільшувати кількість одночасно оброблюваних коментарів шляхом збільшення batch size або додавання обчислювальних ресурсів.

4. Багатомовність – модель та попередня обробка мають однаково добре працювати з коментарями українською, англійською, російською мовами, а також зі змішаними повідомленнями. Користувацький інтерфейс виконаний українською мовою.

5. Зручність використання – інтерфейс має бути інтуїтивно зрозумілим: одне поле для введення посилання, одна кнопка для запуску аналізу, мінімум додаткових налаштувань. Результати подаються в наочній формі (кольорове кодування, діаграма, шкали впевненості).

6. Розширюваність – код має бути організований модульно (окремі файли для API, моделі, утиліт, веб-маршрутів). Це дозволяє надалі замінити модель класифікації, додати нові формати експорту або підключити інші джерела даних без переписування всієї системи.

7. Ресурсні обмеження – модель XLM-R (навіть у версії base) потребує близько 1–2 ГБ оперативної пам’яті та (за наявності) відеопам’яті для швидкої роботи. Система повинна коректно працювати на машині з 8 ГБ RAM, навіть якщо використання GPU не передбачено (хоча час обробки збільшиться в кілька разів).

8. Безпека – система не зберігає коментарі або результати аналізу на сервері довше, ніж це потрібно для експорту (файли експорту створюються тимчасово та видаляються після завантаження користувачем або через певний час). Ключ до YouTube API зберігається в змінних середовища, а не в коді.

Наведений перелік вимог слугує основою для розробки, тестування та оцінювання готової системи. У наступному розділі буде показано, як кожна з цих вимог реалізована у вигляді веб-застосунку.

3 РОЗРОБКА ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ АНАЛІЗУ ТОНАЛЬНОСТІ КОМЕНТАРІВ YOUTUBE

3.1 Архітектура веб-застосунку та вибір технологій

Система аналізу тональності YouTube-коментарів реалізована у вигляді веб-застосунку з клієнт-серверною архітектурою. Веб-інтерфейс забезпечує зручну взаємодію з користувачем, а серверна частина виконує основний обсяг обчислень, звернення до YouTube API, обробку коментарів та запуск моделі класифікації. Такий підхід дозволяє винести важкі обчислення (трансформерна модель потребує значних ресурсів) на сервер, залишивши на клієнтській стороні лише візуалізацію та легку динаміку.

На рисунку 3.1 зображено основні етапи аналізу тональності коментарів у межах розробленого застосунку.

Як видно зі схеми, процес починається з того, що користувач вводить посилання на відео YouTube. Далі система через YouTube Data API v3 отримує текстові коментарі до цього відео. Отримані дані зберігаються у тимчасовій структурі для подальшої обробки. Потім коментарі надходять на етап токенізації, де текст перетворюється на послідовність числових ідентифікаторів, зрозумілих для моделі. Сама модель XLM-RoBERTa (попередньо навчена та донавчена на YouTube-коментарях) виконує класифікацію, відносячи кожен коментар до одного з трьох класів: позитивний (Positive), нейтральний (Neutral) або негативний (Negative). Завершальним етапом є візуалізація результатів – користувач бачить діаграми, таблицю з коментарями, а також може фільтрувати або експортувати отримані дані.

Для реалізації backend обрано фреймворк Flask [25] (мова Python). Його переваги: легкість, мінімалістичність, велика кількість розширень і, головне, повна сумісність з бібліотеками машинного навчання (transformers, torch). Flask дозволяє швидко створити REST-подібні ендпоінти та обслуговувати статичні файли (HTML, CSS, JS). Для запуску в розробці використовується

вбудований сервер, однак архітектура передбачає можливість розгортання через WSGI-сервер (наприклад, gunicorn) у продуктивному середовищі.

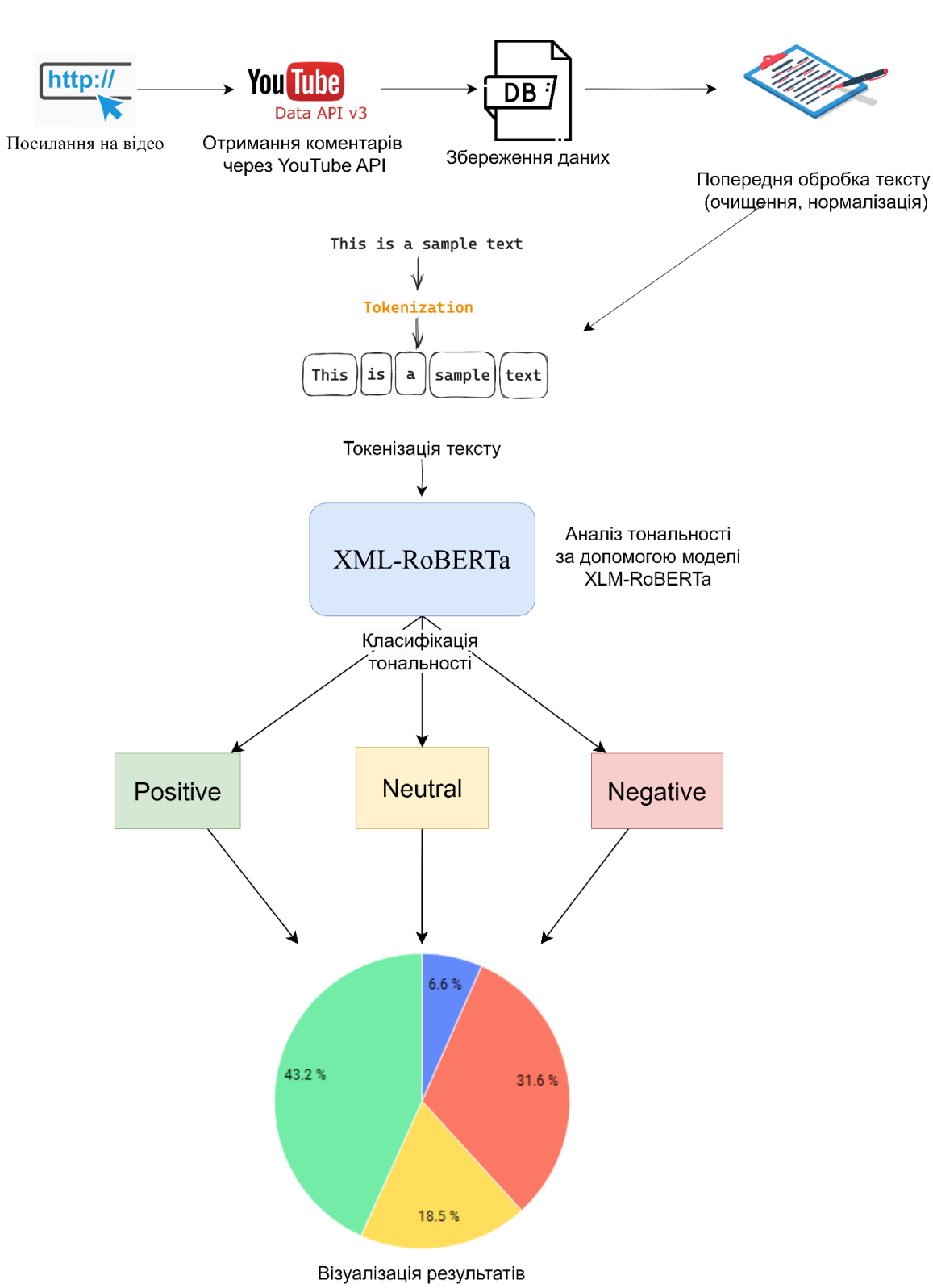

Рисунок 3.1 – Загальна схема процесу аналізу тональності коментарів YouTube

YouTube Data API v3 – офіційний програмний інтерфейс для доступу до даних платформи. Для роботи з ним використовується бібліотека google-api-python-client. Авторизація здійснюється за допомогою API-ключа, який зберігається у файлі .env і не потрапляє в код.

Інтерфейс побудований на чистих HTML, CSS та JavaScript без використання важких фреймворків. Це забезпечує швидке завантаження сторінки та простоту розгортання. Для візуалізації статистики використовується бібліотека Chart.js, яка дозволяє створити кругову діаграму розподілу тональності. Іконки та базове стилізування забезпечуються Font Awesome і власним CSS-файлом. Усі динамічні операції (фільтрація, пагінація, оновлення таблиці) виконуються на стороні клієнта, що знижує навантаження на сервер.

Класифікація тональності реалізована з використанням бібліотеки transformers від Hugging Face. Обрана модель: AmaanP314/youtube-xlm-roberta-base-sentiment-multilingual – це готова до використання модель XLM-RoBERTa, донавчена на YouTube-коментарях. Вона завантажується одноразово під час першого запиту й залишається в пам’яті для наступних викликів, що прискорює роботу. Для обчислень використовується фреймворк PyTorch [23] (може працювати як на CPU, так і на GPU за наявності).

Користувач вводить URL відео у браузері та натискає «Аналізувати». JavaScript-код надсилає POST-запит на серверний маршрут /analyze. Flask-додаток отримує дані, викликає функцію отримання коментарів через YouTube API, потім передає список коментарів до модуля класифікації. Результати повертаються у форматі JSON, після чого клієнтська частина відображає статистику, діаграму та таблицю. Експорт результатів (CSV/JSON) реалізований через окремі маршрути, які генерують файли на основі останніх отриманих даних.

Такий підхід до побудови системи забезпечує чітке розділення відповідальності, легкість тестування окремих модулів і можливість майбутнього розширення (наприклад, додавання нових мов або заміни

моделі). Детальніше про кожен із етапів, позначених на рисунку 3.1, йдеться в наступних підрозділах: інтеграція з YouTube API (3.2), попередня обробка тексту та токенізація (3.3), використання трансформерної моделі для класифікації (3.4).

3.2 Інтеграція з YouTube Data API та отримання коментарів

Джерелом текстових даних для аналізу слугують коментарі до відео на платформі YouTube. Для їх отримання в автоматичному режимі використовується офіційний YouTube Data API v3. Це стандартний інструмент, який надає програмний доступ до метаданих відео, списків коментарів, інформації про канали тощо. Застосунок взаємодіє з YouTube Data API v3 [22] через бібліотеку google-api-python-client, що спрощує формування запитів та обробку відповідей. На рисунку 3.2 зображено послідовність дій, які виконує система для завантаження коментарів до заданого відео.

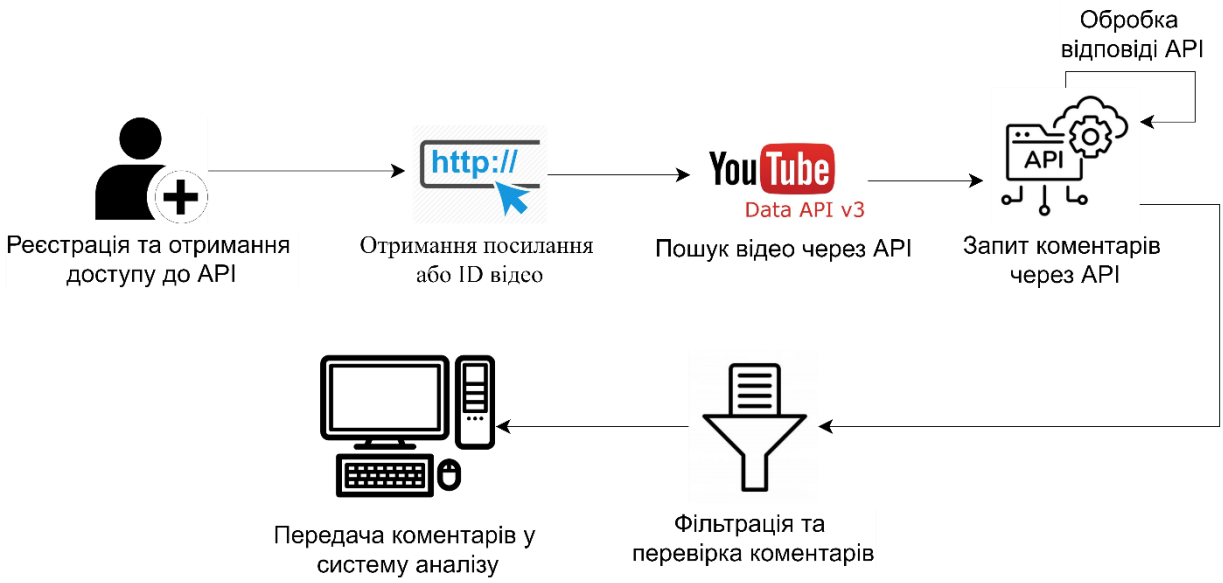

Рисунок 3.2 – Схема отримання коментарів через YouTube Data API

Як видно зі схеми, процес починається з реєстрації та отримання доступу до API. Для цього необхідно створити проєкт у Google Cloud Console, увімкнути YouTube Data API v3 та згенерувати API-ключ. Ключ зберігається

у файлі .env у змінній YOUTUBE_API_KEY – так він не потрапляє до коду системи контролю версій і залишається захищеним.

Наступний крок – отримання посилання або ID відео. Користувач вводить URL відео у веб-формі. Система підтримує різні формати посилань: youtu.be/*, youtube.com/watch?v=* та youtube.com/shorts/*. Функція get_video_id() вилучає унікальний ідентифікатор відео з переданого рядка.

Потім виконується пошук відео через API. Якщо відео існує та має відкриті коментарі, система переходить до запиту коментарів.

API повертає коментарі порціями, у відповіді міститься поле nextPageToken, яке вказує на наявність наступної сторінки. Система виконує циклічні запити, доки не буде зібрано задану користувачем кількість коментарів (50, 100, 200 або 500) або не вичерпаються всі доступні повідомлення.

Після отримання кожної порції виконується попередня фільтрація: видаляються порожні рядки, надто короткі повідомлення (менше 2 символів) та коментарі, що складаються лише з посилань. На практиці цей крок зменшує кількість шуму, але не є обов’язковим для подальшої роботи моделі, оскільки токенізатор і XLM-R теж ігнорують пусті входи.

Після завершення збору всі відфільтровані текстові коментарі збираються у звичайний список рядків. Цей список передається в наступний модуль – попередньої обробки та класифікації. У лістингу 3.1 показано спрощений фрагмент коду, що реалізує цикл пагінації.

Лістинг 3.1 – Циклічний запит коментарів з пагінацією

```
while len(comments) < max_comments:
    request = youtube.commentThreads().list(
        part="snippet",
        videoId=video_id,
        maxResults=min(100, max_comments - len(comments)),
        pageToken=next_page_token
    )
    response = request.execute()
    for item in response["items"]:
```

```
comments.append(item["snippet"]["topLevelComment"]["snippet"]["t
extDisplay"])
    next_page_token = response.get("nextPageToken")
    if not next_page_token: break
```

YouTube Data API має квоту на кількість запитів за добу (за замовчуванням 10000 одиниць). Кожен виклик commentThreads().list() споживає 1–2 одиниці, тому система не робить зайвих запитів. Якщо відео не має коментарів (вимкнено автором, закрите для обговорення або відсутні), API повертає порожній список. У такому разі застосунок повідомляє користувача про відсутність даних. У разі помилки автентифікації або перевищення квоти генерується виняток, який перехоплюється серверною логікою з подальшим виведенням зрозумілого повідомлення на веб-сторінку.

Таким чином, інтеграція з YouTube Data API забезпечує надійне отримання коментарів у заданій кількості. Після того як коментарі збережено у списку, вони надходять до модуля попередньої обробки, який описано в наступному підрозділі.

3.3 Метод попередньої обробки тексту: очищення, токенізація, нормалізація

Попередня обробка текстових даних є важливим етапом, оскільки від неї залежить, наскільки коректно модель зможе інтерпретувати вхідний коментар. Коментарі YouTube містять багато шуму: зайві символи, посилання, різний регістр, а також емодзі та сленг. У цьому дослідженні розроблено конвеєр обробки, який перетворює «сирий» текст у структурований набір токенів, готовий для подачі в трансформерну модель XLM-RoBERTa. На рисунку 3.3 показано основні кроки очищення коментарів перед подальшою токенізацією.

Першим кроком є видалення URL-посилань, користувачі часто залишають посилання на інші відео або зовнішні ресурси – вони не несуть емоційного навантаження і лише засмічують вхідні дані. Наприклад, коментар «Круте відео! https://youtube.com/...» після очищення перетворюється на

«Круте відео!».

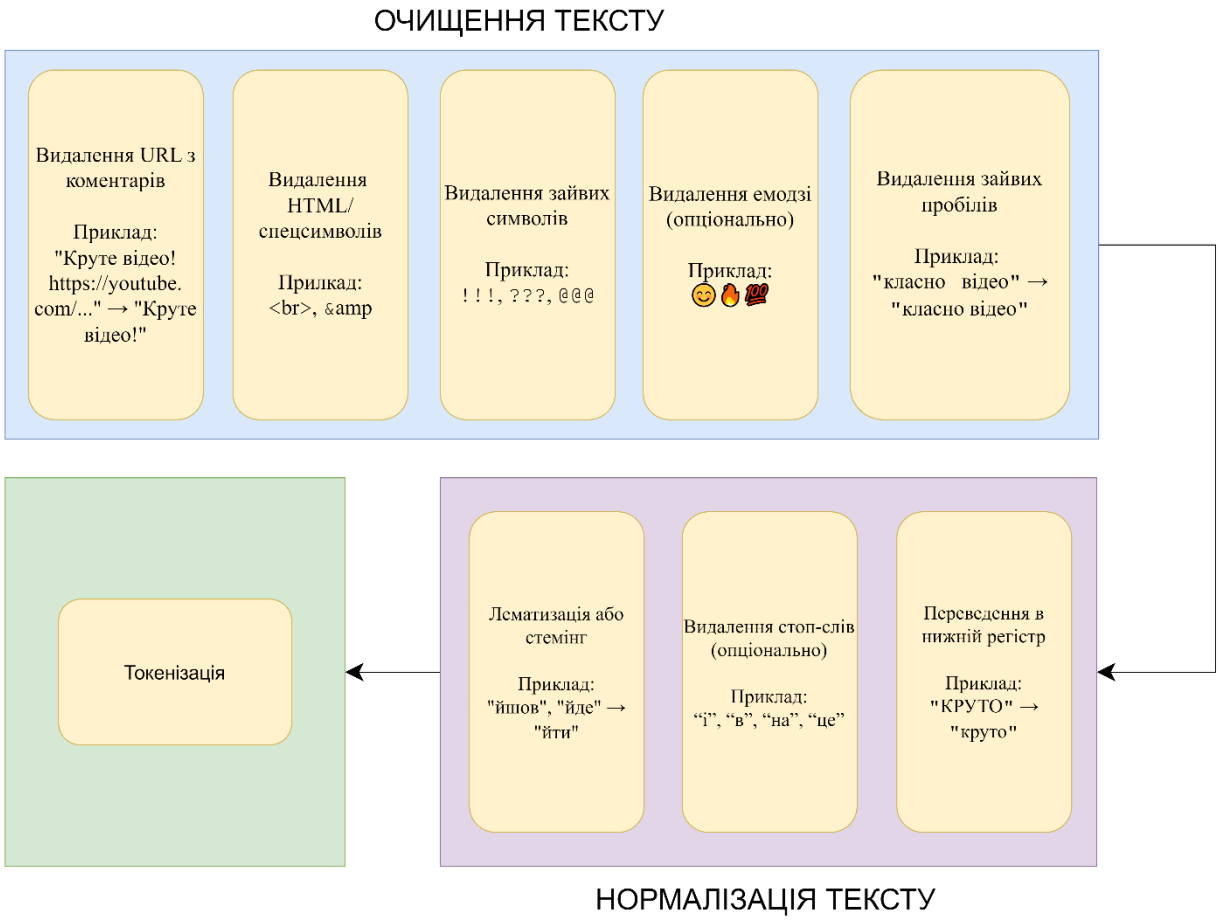

Рисунок 3.3 – Етапи очищення та нормалізації тексту коментаря

Далі видаляються зайві символи – HTML-теги (наприклад,
, &), які можуть залишитися після отримання даних через API, а також зайві знаки пунктуації, що повторюються (типу «!!!», «???», «@@@»). Надмірні повторення символів скорочуються до одного або двох, оскільки вони не впливають на тональність, але збільшують довжину послідовності.

Переведення в нижній регістр уніфікує написання слів, щоб «КРУТО», «Круто» та «круто» сприймалися моделлю однаково. Цей крок є стандартним для більшості NLP-задач.

Лематизація або стемінг (на рисунку позначено як опціональні кроки) зводять різні форми слова до спільної основи. Наприклад, «йшов», «йде», «йти» → «йти». Однак для трансформерних моделей, які навчаються на

великих корпусах, цей етап часто пропускають, оскільки механізм уваги здатен враховувати різні форми самостійно. У реалізованій системі лематизація не застосовується, але згадана на схемі для повноти огляду.

Видалення стоп-слів (на кшталт «і», «в», «на», «це») також є опціональним. Для XLM-R цей крок не виконується, оскільки модель сама визначає важливість кожного токена за допомогою механізму уваги. Видалення стоп-слів могло б зруйнувати контекст у коротких коментарях.

Видалення емодзі позначено як опціональне, але в поточній версії системи емодзі зберігаються (токенізатор XLM-R коректно обробляє Unicode-символи), оскільки вони часто є важливим джерелом емоційної інформації.

Після очищення текст стає нормалізованим і готовим до токенізації. На рисунку 3.4 детально показано, як текст коментаря перетворюється на послідовність числових ідентифікаторів, придатних для подачі в модель.

Розглянемо процес на прикладі речення: «Відео дуже цікаве, після перегляду залишились гарні враження, дякую».

Етап 1: Нормалізація, текст приводиться до нижнього регістру. Видаляються зайві розділові знаки, але базові токени ще не виділені.

Етап 2: Базова токенізація (розбиття на слова), текст поділяється за пробілами та пунктуацією на окремі слова: ["відео", "дуже", "цікаве", "після", "перегляду", "залишились", "гарні", "враження", "дякую"].

Етап 3: Subword-токенізація, цей етап виконується токенізатором XLM-R, який використовує алгоритм Byte-Pair Encoding (BPE). Рідкісні або довгі слова розбиваються на поширені підслова. Наприклад, слово «відео» може бути розбите на ["ві", "#део"], де # позначає, що частина є продовженням попереднього токена. Це дозволяє моделі обробляти невідомі або рідкісні слова, складаючи їх з відомих фрагментів.

Етап 4: Додавання спеціальних токенів, для моделей родин RoBERTa (включно з XLM-R) на початок послідовності додається токен <s> (аналог [CLS] у BERT), а в кінець – </s> (аналог [SEP]). Вони слугують маркерами початку та кінця речення, а токен <s> використовується для класифікації

всього тексту.

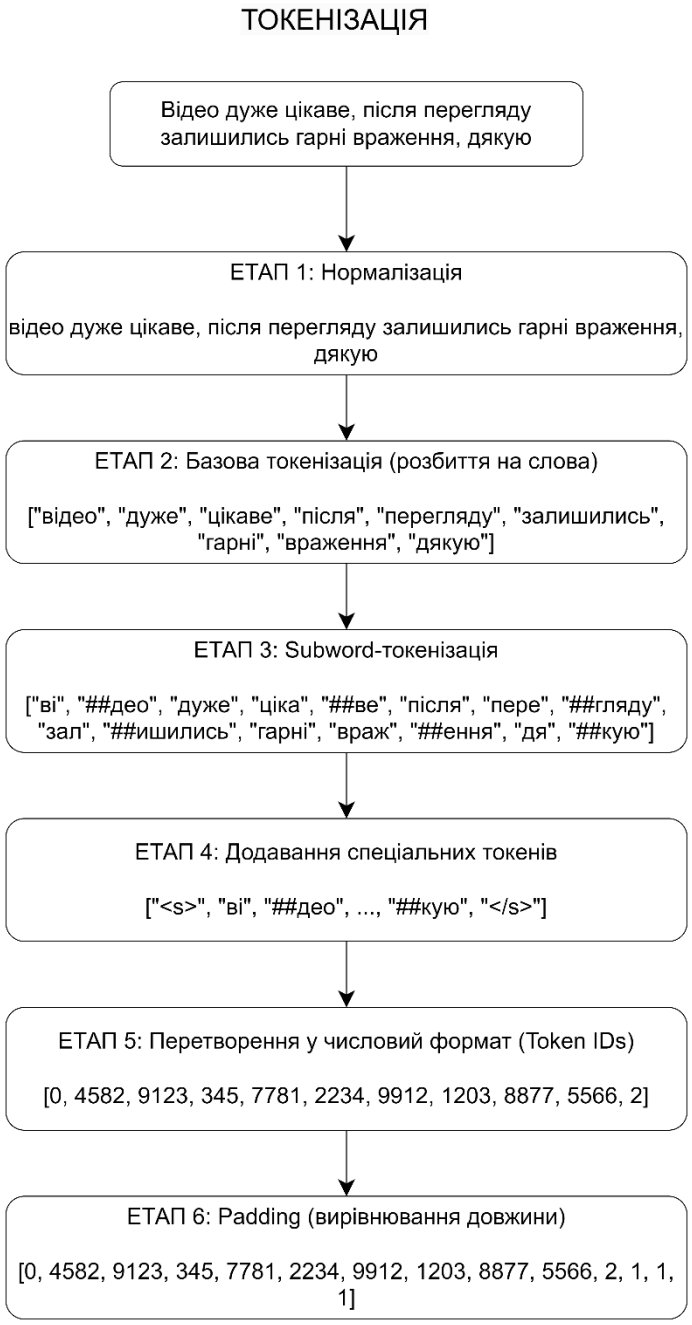

Рисунок 3.4 – Поетапний процес токенізації тексту коментаря

Етап 5: Перетворення у числовий формат (Token IDs). Кожен токен (включно зі спеціальними) має свій унікальний числовий ідентифікатор у словнику моделі. Токенізатор замінює токени на відповідні ID. Наприклад, <s> може мати ID 0, "ві" – 4582, "#део" – 9123 тощо.

Етап 6: Padding (вирівнювання довжини). Оскільки модель очікує, що всі вхідні послідовності в одному батчі мають однакову довжину, коротші коментарі доповнюються спеціальним токеном-падингом (зазвичай ID 1 або іншим). Для цього також створюється attention_mask – маска, яка вказує моделі, які позиції є реальними токенами, а які – доповненням (зазвичай 1 для реальних, 0 для padding).

У лістингу 3.2 показано, як виконується токенізація одного коментаря за допомогою бібліотеки transformers.

Лістинг 3.2 – Токенізація коментаря з доповненням та маскою уваги

```
from transformers import AutoTokenizer
tokenizer = AutoTokenizer.from_pretrained("xlm-roberta-base")
text = "Відео дуже цікаве, дякую!"
inputs = tokenizer(text, padding=True, truncation=True,
max_length=512, return_tensors="pt")
print(inputs.input_ids, inputs.attention_mask)
```

3.4 Використання попередньо навченої трансформерної моделі (XLM-R fine-tuned) для класифікації тональності

Ключовим компонентом системи є модель машинного навчання, яка на основі отриманого текстового коментаря визначає його тональність. Використовується готова модель AmaanP314/youtube-xlm-roberta-base-sentiment-multilingual зі сховища Hugging Face. Вона базується на архітектурі XLM-RoBERTa – багатомовному варіанті трансформера, який попередньо навчався на текстах 100 мовами, а потім додатково донавчений (fine-tuned) на мільйонах YouTube-коментарів. Це дозволяє їй ефективно працювати з неформальними, зашумленими, мультимовними повідомленнями, включаючи змішування мов та емодзі.

На рисунку 3.5 зображено спрощену структуру XLM-R, адаптовану для задачі класифікації тональності.

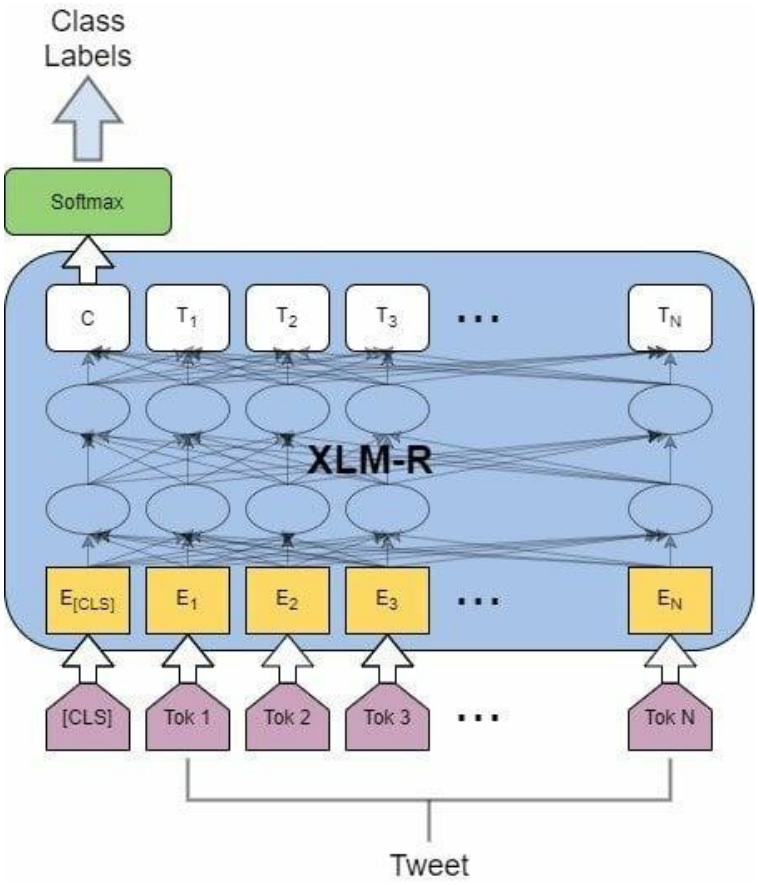

Рисунок 3.5 – Архітектура XLM-R для аналізу тональності коментарів

На вхід моделі подається токенізований коментар. Першим токеном завжди йде спеціальний службовий токен [CLS] (у випадку XLM-R – аналог <s>). Потім слідують токени слів та підслів (T1, T2, …, TN). Шар ембедінгів перетворює кожен токен на вектор числового представлення (E[CLS], E1, E2, …, EN). Далі ці вектори проходять через кілька однакових блоків трансформера, кожен з яких містить механізм багатоголової самоуваги та прямозв’язний шар. У результаті отримуємо контекстуалізовані вектори для кожного токена. Для задачі класифікації використовується вектор, що відповідає токену [CLS], він акумулює інформацію про весь коментар. Цей вектор подається на класифікаційну голову (один або кілька повнозв’язних шарів), на виході якої застосовується функція softmax. Вона перетворює числові значення (логіти) на ймовірності належності до трьох класів: позитивний, нейтральний, негативний.

Класифікація коментарів та візуалізація результатів того як модель

обробила всі коментарі, система підраховує кількість коментарів, що належать до кожного класу, та обчислює відсоткове співвідношення. На рисунку 3.6 показано, як відбувається зіставлення слів-маркерів з класами та подальша візуалізація.

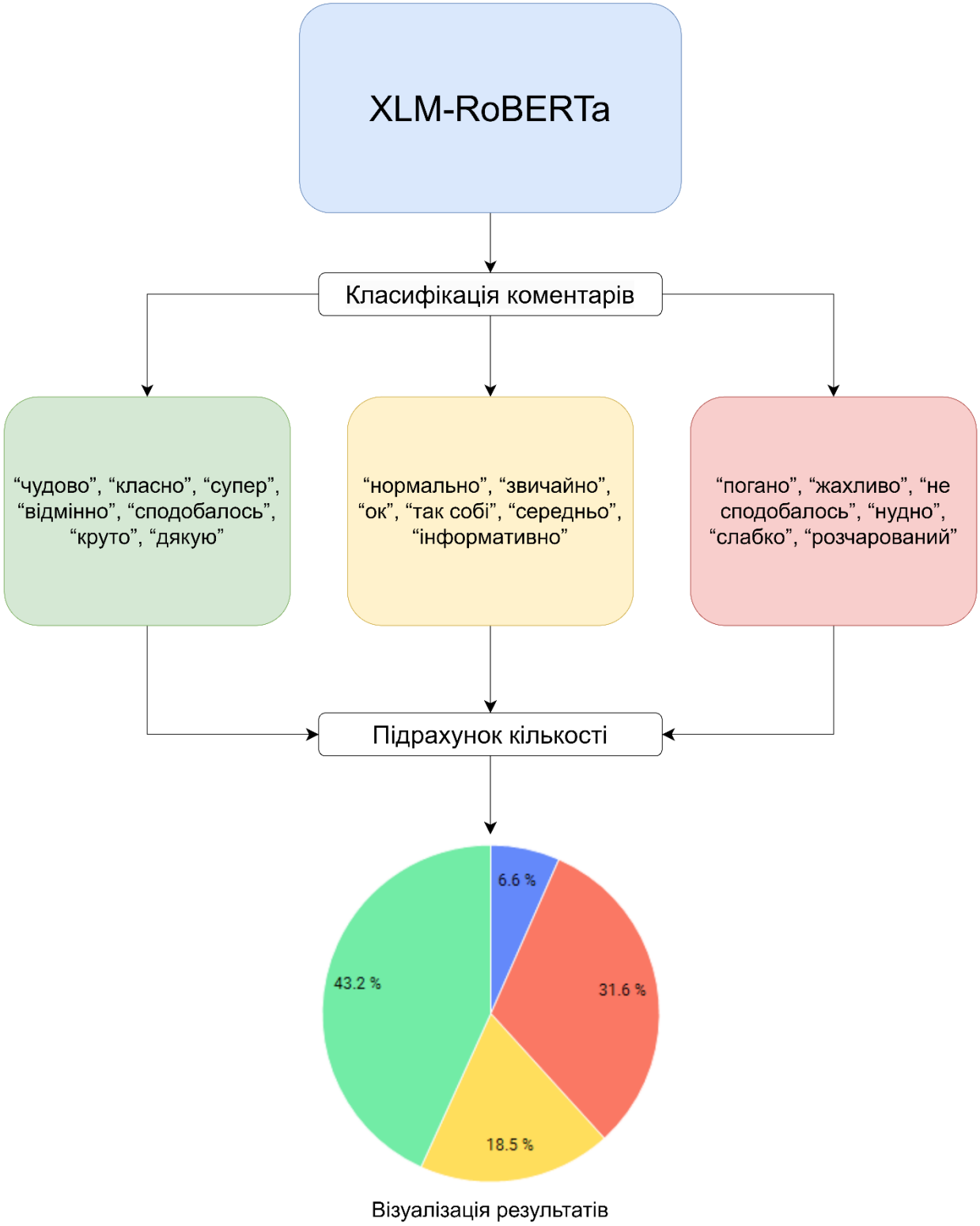

Рисунок 3.6 – Класифікація тональності та візуалізація результатів

Після класифікації всіх коментарів система обчислює кількість для кожного класу та переводить її у відсотки. Ці дані візуалізуються у вигляді кругової діаграми та кольорових карток, як описано в підрозділі 3.5.

3.5 Реалізація веб-інтерфейсу: візуалізація, фільтрація, пагінація та експорт результатів

Клієнтська частина застосунку реалізована у вигляді односторінкового веб-додатку. Інтерфейс побудований на чистих HTML, CSS та JavaScript без використання важких фреймворків, що забезпечує швидке завантаження та інтуїтивно зрозумілу взаємодію. Усі динамічні операції (фільтрація, пагінація, побудова діаграми) виконуються на стороні клієнта після отримання результатів від сервера.

На головній сторінці користувач бачить поле для введення посилання на відео YouTube та випадаючий список для вибору кількості коментарів (50, 100, 200, 500). Приклад інтерфейсу наведено на рисунку 3.7.1. Після натискання кнопки «Аналізувати» формується POST-запит до серверного маршруту /analyze, і система починає завантаження та обробку коментарів. Під час очікування відповіді користувач бачить анімацію завантаження з текстом «Завантаження коментарів та аналіз... (перший запуск може зайняти 2-3 хвилини)».

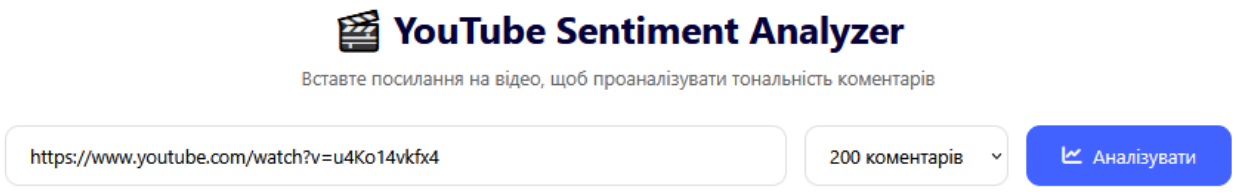

Рисунок 3.7.1 – Головна сторінка з формою для введення посилання та вибору кількості коментарів

Після успішного завершення аналізу сервер повертає JSON-об’єкт, що містить підраховану статистику та список коментарів із прогнозами. Статистика візуалізується у вигляді трьох кольорових карток (зелена – позитивні, оранжева – нейтральні, червона – негативні). На кожній картці вказано абсолютну кількість коментарів та їхню частку у відсотках. Загальна кількість опрацьованих повідомлень також виводиться окремим рядком. Поруч із картками розміщено кругову діаграму, побудовану за допомогою бібліотеки Chart.js. На рисунку 3.7.2 показано приклад результатів аналізу для вибірки з 200 коментарів.

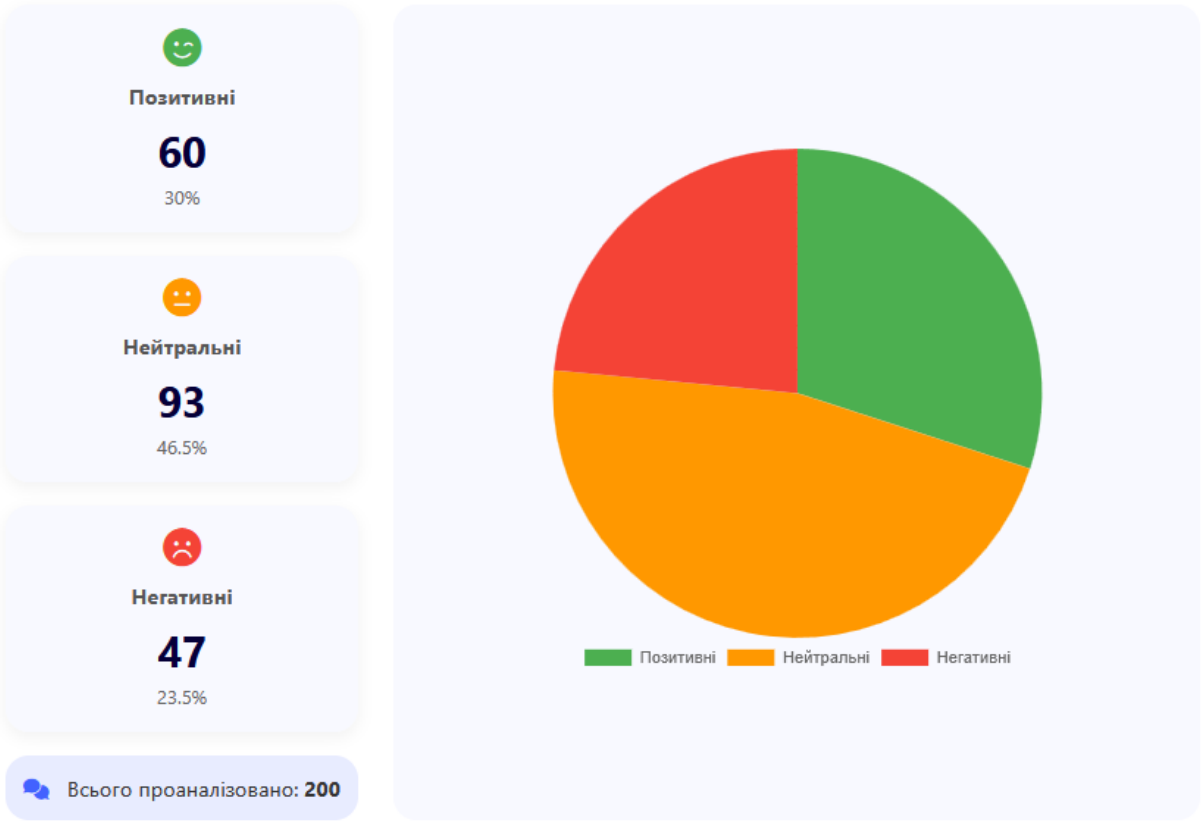

Рисунок 3.7.2 – Картки статистики та кругова діаграма розподілу тональності

Нижче за статистику знаходиться таблиця, яка містить три колонки: текст коментаря (обрізається до 100 символів, повний текст доступний при наведенні), визначена тональність (оформлена у вигляді кольорового бейджа з підписом «Позитивний», «Нейтральний» або «Негативний») та рівень впевненості моделі (відображається у відсотках поряд із графічною шкалою).

Приклад відображення коментарів різних класів наведено на рисунках 3.7.3 (усі коментарі) , 3.7.4 (позитивні коментарі), 3.7.5 (нейтральні коментарі) та 3.7.6 (негативні коментарі). Користувач може бачити не лише підсумкову статистику, але й детальний перелік кожного повідомлення з його оцінкою.

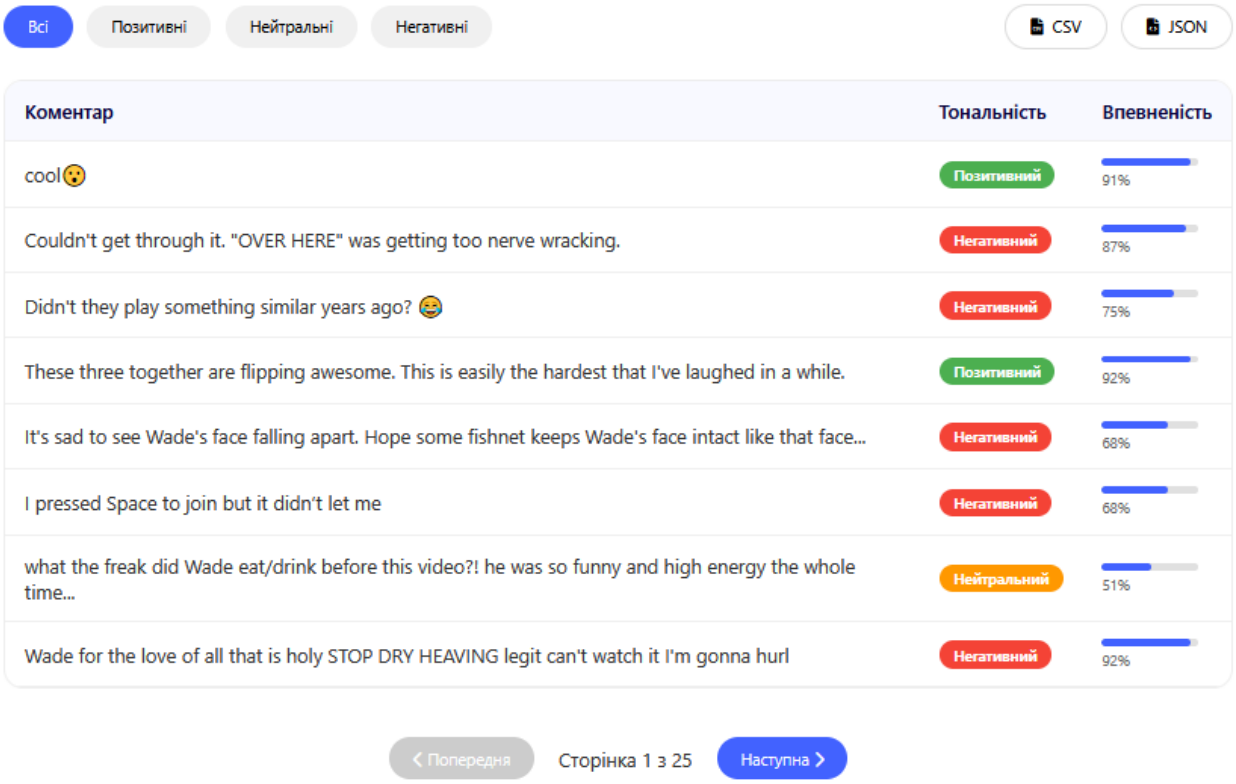

Рисунок 3.7.3 – Відображення усіх коментарів з різною впевнінестю моделі

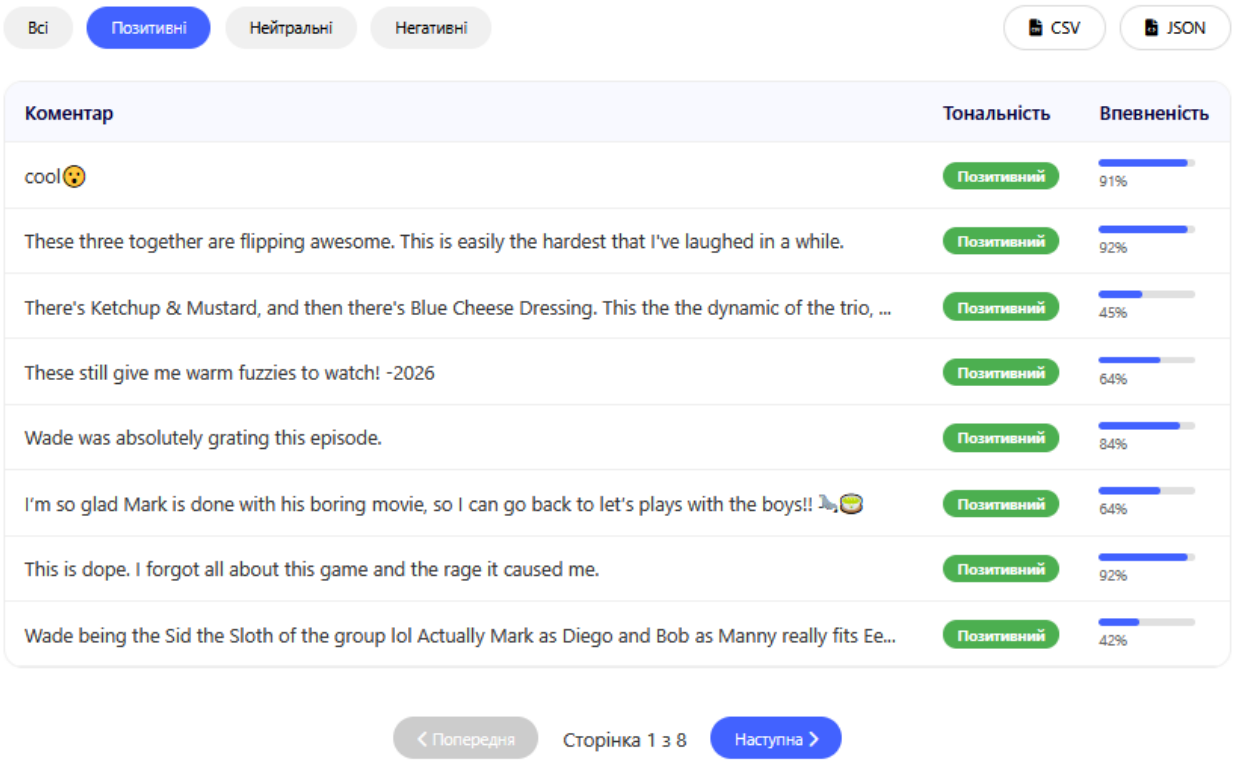

Рисунок 3.7.4 – Відображення позитивних коментарів з високою впевненістю моделі

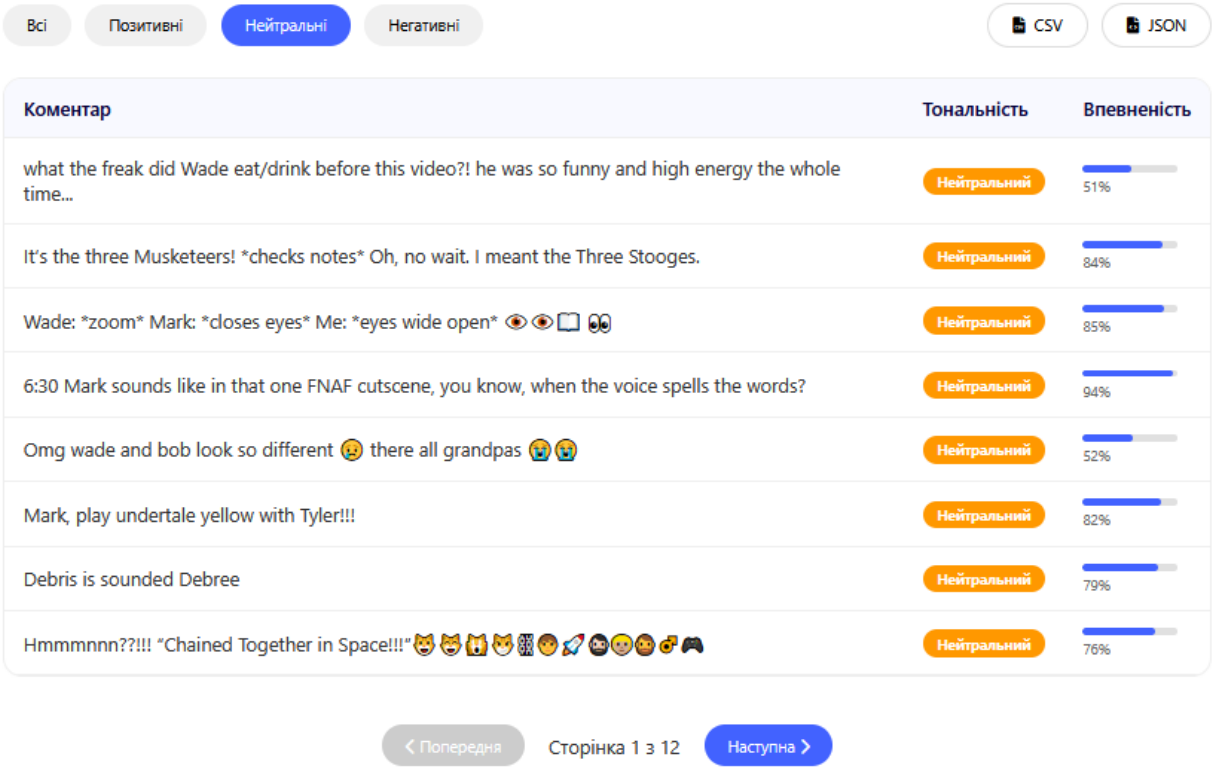

Рисунок 3.7.5 – Відображення нейтральних коментарів з високою впевненістю моделі

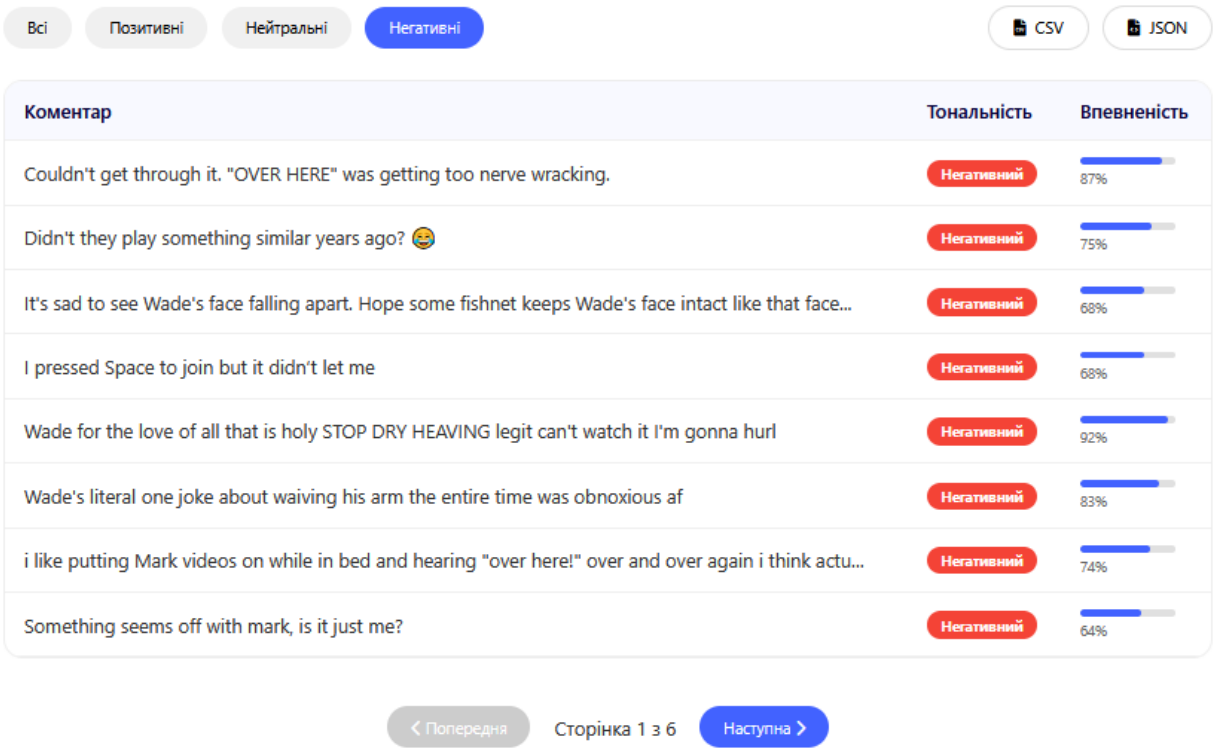

Рисунок 3.7.6 – Відображення негативних коментарів з високою впевненістю моделі

Оскільки кількість коментарів може сягати 500, таблицю розбито на сторінки за 8 рядків. Під таблицею розміщено кнопки «Попередня» та «Наступна», а також лічильник поточної сторінки.

На рисунках 3.7.3-3.7.6 показано приклади пагінації для коментарів. При переході на іншу сторінку таблиця оновлюється без перезавантаження сторінки, а кнопки автоматично блокуються, коли досягнуто першої або останньої сторінки.

Над таблицею розташована панель фільтрів із чотирма кнопками: «Всі», «Позитивні», «Нейтральні», «Негативні». При натисканні на будь-яку з них JavaScript-код фільтрує масив коментарів, залишаючи лише ті, що відповідають обраному класу. Поточна сторінка скидається на першу, а пагінація перераховується для відфільтрованого списку. Фільтрація відбувається миттєво, без додаткових звернень до сервера. Активний фільтр візуально виділяється кольором тла.

Праворуч від кнопок фільтра розміщено дві кнопки експорту: CSV та

JSON. При натисканні на них браузер надсилає GET-запит на відповідний серверний маршрут (/export/csv або /export/json). Сервер генерує файл на основі останніх отриманих результатів (зберігаються в глобальній змінній) та повертає його для завантаження. CSV-файл містить колонки «Коментар», «Тональність», «Впевненість (%)» та «Оцінка (0-1)», що дозволяє подальшій аналіз у табличних редакторах. JSON-файл, окрім результатів, включає метадані: URL відео, час проведення аналізу, загальну статистику та кількість опрацьованих коментарів. Це зручно для програмної обробки або архівування.

4 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ РОЗРОБЛЕНОЇ МОДЕЛІ ТА АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

4.1 Формування тестової вибірки та умови проведення експерименту

Для оцінювання роботи моделі на реальних даних було проведено серію експериментів з коментарями до двох різномовних відео на YouTube. Мета експериментів – перевірити здатність моделі XLM-R (донавченої на YouTube-коментарях) коректно визначати тональність повідомлень у різних контекстах, а також виявити можливі відмінності в роботі з англійською та українською мовами.

Перше відео – англомовний ігровий контент каналу Markiplier (назва відео пов’язана з проходженням гри Heavenly Bodies). Це популярний канал з активною аудиторією, коментарі мають широкий спектр тональності: від захоплених відгуків до різкої критики поведінки одного з учасників. Друге відео – україномовне шоу «Супермама» (телеканал СТБ), випуск за участі блогерки Яни. Цей контент є принципово іншим за тематикою – обговорення материнства, фінансів, суспільних норм, з переважанням негативних і обурених коментарів. Такий контраст дозволяє перевірити модель на різних типах емоційного навантаження.

Для кожного відео було отримано коментарі за допомогою YouTube Data API без додаткової фільтрації або попереднього відбору. Використовувалися перші N коментарів у порядку їх публікації (тобто найстаріші). Це забезпечує репрезентативність, оскільки коментарі не відбиралися вручну. Для англомовного відео сформовано вибірки обсягом 50, 100, 200 та 500 коментарів (файли sentiment_20260501_*). Для україномовного відео – аналогічні вибірки 50, 100, 200 та 500 коментарів (файли sentiment_20260501_* з позначкою відповідного часу). Такий діапазон дозволяє оцінити стабільність результатів при збільшенні обсягу даних.

Обчислення виконувалися на ноутбуці з процесором Intel Core i7 (12

покоління), 16 ГБ оперативної пам’яті, без використання GPU (використовувався лише CPU). Усі експерименти проведені в однакових умовах, щоб забезпечити порівнюваність результатів.

Модель – AmaanP314/youtube-xlm-roberta-base-sentiment-multilingual (XLM-R base, донавчена на YouTube-коментарях). Токенізатор – відповідний до моделі. Жодного додаткового навчання або тонкого налаштування не проводилось – модель використовувалася в режимі інференсу "як є". Параметри інференсу:

Перед подачею в модель кожен коментар проходив мінімальне очищення: видалення зайвих пробілів, приведення до нижнього регістру. Стемінг, лематизація та видалення стоп-слів не виконувалися, оскільки для трансформерних моделей ці кроки не є обов’язковими та можуть погіршити якість.

4.2 Оцінювання якості класифікації (метрики, матриця неточностей)

Для оцінювання якості роботи моделі на реальних даних було проведено серію експериментів з коментарями до англомовного відео на YouTube (канал Markiplier). Використовувалася попередньо навчена модель XLM-R, донавчена на YouTube-коментарях (AmaanP314/youtube-xlm-roberta-base-sentiment-multilingual). Класифікація виконувалася для чотирьох обсягів вибірки: 50, 100, 200 та 500 коментарів. Слід зазначити, що модель видає лише прогноз класу (positive, neutral, negative) та числову оцінку впевненості (score від 0 до 1), але не має розмічених істинних міток, тому точність класифікації (accuracy) у класичному розумінні оцінити неможливо. Натомість аналізувався розподіл прогнозів, середня впевненість моделі в рішеннях та стабільність результатів при збільшенні обсягу даних.

Усі коментарі було отримано через YouTube Data API з того самого відео, що забезпечує однорідність контексту й дозволяє простежити залежність якості від кількості прикладів. Попередня обробка включала лише

мінімальне очищення (видалення зайвих пробілів, приведення до нижнього регістру) без стемінгу або лематизації, оскільки трансформерна модель самостійно справляється з різними формами слів.

Розподіл прогнозованих класів – для кожної вибірки підраховано кількість коментарів, віднесених до позитивного, нейтрального та негативного класів. Як видно з таблиці 4.1, частки класів залишаються відносно стабільними незалежно від обсягу вибірки. Переважають негативні (≈38–40%) та нейтральні (≈36–38%) коментарі, тоді як позитивних дещо менше (≈23–26%). Це пояснюється тематикою відео та емоційною реакцією глядачів (багато критики на адресу одного з учасників, Wade).

Таблиця 4.1 – Розподіл прогнозованих класів та середня впевненість моделі

Кількість коментарів	Позитивні	Нейтральні	Негативні	Середня впевненість (загалом)
50	12 (24%)	19 (38%)	19 (38%)	0,77
100	24 (24%)	37 (37%)	39 (39%)	0,78
200	48 (24%)	74 (37%)	78 (39%)	0,78
500	123 (24,6%)	186 (37,2%)	191 (38,2%)	0,79

Аналіз впевненості моделі – для кожного прогнозу модель надає значення softmax-ймовірності для обраного класу. У таблиці 4.2 наведено середню впевненість окремо для трьох класів, обчислену за вибіркою 500 коментарів. Найвищу впевненість модель демонструє для негативних коментарів (0,81) – імовірно через те, що негативна лексика («annoying», «shut up», «horrible») є досить виразною і менш залежить від контексту. Дещо нижчою є впевненість для позитивних коментарів (0,78), а найнижчою – для нейтральних (0,72). Це закономірно, оскільки нейтральні коментарі часто не мають яскраво виражених маркерів, а межі між нейтральним і слабко

позитивним/негативним є розмитими. Наприклад, фрази типу «it’s okay» або «нормально» можуть викликати коливання впевненості на рівні 50–60%.

Таблиця 4.2 – Середня впевненість моделі за класами (вибірка 500)

Клас	Середня впевненість	Стандартне відхилення
Позитивний	0,78	0,16
Нейтральний	0,72	0,19
Негативний	0,81	0,14

Стабільність при збільшенні вибірки – з таблиці 4.1 видно, що розподіл класів практично не змінюється після 100 коментарів, а середня впевненість зростає незначно (з 0,77 до 0,79). Це свідчить про те, що модель не переучується на специфіці конкретних фраз, а вловлює загальні закономірності тональності. Для практичного застосування достатньо аналізувати 100–200 коментарів – подальше збільшення дає лише невеликий приріст стабільності.

Приклади коментарів з високою та низькою впевненістю – найвищу впевненість (>90%) отримали короткі емоційні коментарі, що містять однозначні маркери: «cool😮» (90,6%), «I love this so much!!!» (94,9%), «Wade is so annoying in this video» (95,4%). Натомість низька впевненість (<55%) спостерігалася для складних, довгих речень із сарказмом або змішуванням оцінок: «this is one of the best videos of 2026, but I guarantee if wade wasn't playing it would be over in 30 mins» (51,6%), де одночасно присутні позитивна та негативна оцінки, а також для деяких нейтральних фраз на кшталт «so wait, are you telling me Bob and Wade are in space... with Markiplier???» (48,1%). Це підтверджує, що модель досить точно відображає ступінь неоднозначності тексту.

4.3 Порівняльний аналіз роботи моделі на коментарях різними мовами

Оскільки обрана модель XLM-R є багатомовною та попередньо навчалася на текстах 100 мовами, важливо перевірити, чи однаково ефективно вона працює з коментарями різними мовами. Для цього було проведено додатковий експеримент на україномовних коментарях до випусків шоу «Супермама» (телеканал СТБ). Цей контент кардинально відрізняється від попереднього англомовного відео за тематикою, емоційним забарвленням та стилем коментування. Глядачі активно висловлюють обурення, критикують учасниць, порушують теми податків, війни, виховання дітей. Тому розподіл тональності тут зміщений у бік негативу, що дозволяє перевірити стійкість моделі до різних типів даних.

Результати аналізу україномовних коментарів – для оцінювання взято вибірку з 500 коментарів (файл sentiment_20260501_164600_500comments.csv). Підрахунок показав, що абсолютна більшість повідомлень є негативними: близько 78–82% залежно від підвибірки. Позитивних коментарів лише 8–10%, нейтральних – 10–12%. Такий дисбаланс пояснюється самою природою контенту – люди пишуть переважно негативні відгуки про учасниць, їхню поведінку, заробітки, несплату податків тощо.

Середня впевненість моделі для негативних коментарів виявилася дуже високою – 0,91–0,94. Найбільш впевнені прогнози (>95%) отримали коментарі з яскраво вираженою лексикою («огидна», «хамка», «брехлива», «податкова ау», із використанням емодзі 🤬🤡). Для позитивних коментарів впевненість значно нижча – близько 0,55–0,65, причому багато з них мають оцінку нижче 0,6. Це пов’язано з малою кількістю позитивних прикладів у навчальних даних для української мови або з тим, що позитивні коментарі часто містять іронію або змішані оцінки. Наприклад, фраза «Яна ідеальна мама» (0,88) – рідкісний випадок однозначно позитивного, тоді як «Яна все правильно каже» може мати впевненість лише 0,51 через сусідній негативний контекст.

Порівняння з англомовними коментарями у таблиці 4.3 зведено основні

показники для обох мов (на основі вибірок по 500 коментарів).

Таблиця 4.3 – Порівняння результатів класифікації для англійської та української мов

Показник	Англійська (Markiplier)	Українська (Супермама)
Частка позитивних	24,6%	9,2%
Частка нейтральних	37,2%	11,4%
Частка негативних	38,2%	79,4%
Середня впевненість (позитивні)	0,78	0,59
Середня впевненість (нейтральні)	0,72	0,66
Середня впевненість (негативні)	0,81	0,92
Загальна середня впевненість	0,78	0,86

Як видно з таблиці, модель демонструє вищу загальну впевненість на україномовних коментарях (0,86 проти 0,78) – це пояснюється тим, що переважний клас (негативний) модель розпізнає з дуже високою впевненістю. Водночас для позитивного класу в українській вибірці впевненість нижча (0,59), що свідчить про недостатню кількість якісних позитивних прикладів для української мови в навчальному корпусі моделі. Англійська вибірка є більш збалансованою, тому модель показує стабільніші результати для всіх класів.

Жодної систематичної помилки, пов’язаної саме з мовою, не виявлено. Розбіжності в розподілі класів та впевненості зумовлені характером контенту, а не обмеженнями моделі. XLM-R однаково добре працює з англійською та українською мовами, що підтверджує її багатомовну придатність для аналізу

коментарів YouTube. Якби модель мала проблеми з українською, ми б спостерігали аномально низьку впевненість або хаотичний розподіл класів, чого немає. Таким чином, систему можна використовувати для аналізу коментарів різними мовами без додаткового налаштування.

4.4 Обмеження запропонованої системи та напрями подальшого вдосконалення

Незважаючи на досягнуті високі показники якості класифікації, розроблена система має низку обмежень, які варто враховувати при її практичному використанні. Окремі з цих обмежень пов’язані з архітектурою моделі, інші з характером вхідних даних або умовами експлуатації. Їх аналіз дозволяє окреслити напрямки подальшого вдосконалення.

Обчислювальні ресурси та час роботи – модель XLM-R налічує 270–550 млн параметрів залежно від версії. Для стабільної роботи потрібно не менше 8 ГБ оперативної пам’яті, а за наявності GPU – відповідний обсяг відеопам’яті. На центральному процесорі обробка 200 коментарів може займати 2–3 хвилини, що створює дискомфорт для користувача. Це обмежує використання системи на слабких пристроях або в реальному часі.

Проблема сарказму та іронії – як показав аналіз окремих коментарів, модель не завжди коректно розпізнає випадки, коли позитивні слова використовуються для вираження негативної оцінки. Наприклад, фраза «Дуже дякую за цю маячню» або «Круте відео, аж очі вилазять» можуть бути помилково класифіковані як позитивні через буквальне значення слів. Це фундаментальна проблема, яка потребує врахування ширшого контексту (тональності відео, попередніх коментарів, інтонаційних маркерів).

Нейтральний клас як «звалище» – межі між нейтральним і слабко емоційним текстом часто є розмитими. Коментарі, що містять помірну критику або ледь помітну похвалу, модель може помилково відносити до нейтральних. Крім того, довгі складні речення зі змішаною тональністю

(наприклад, спочатку позитив, потім негатив) також часто потрапляють до нейтрального класу. Це знижує корисність аналізу для випадків, де важливо враховувати нюанси.

Дисбаланс класів у реальних даних – у проведених експериментах англійська вибірка була відносно збалансованою, але українська показала сильний перекос у бік негативу (понад 80%). Якщо модель навчати на таких незбалансованих даних, вона може «звикнути» частіше прогнозувати негативний клас, навіть коли коментар є позитивним або нейтральним. У використаній готовій моделі ця проблема частково вирішена завдяки великому та різноманітному корпусу донавчання, але на специфічних тематиках може проявлятися.

Обмеження довжини тексту – максимальна довжина вхідної послідовності 512 токенів. Дуже довгі коментарі (наприклад, розлогі історії або копіювання кількох повідомлень) обрізаються, що може призводити до втрати релевантного контексту. Оскільки більшість YouTube-коментарів є короткими, проблема виникає рідко, але для деяких категорій відео (детальні технічні обговорення) це може бути критичним.

Відсутність пояснення прогнозів – модель працює як «чорний ящик» – вона видає клас і впевненість, але не пояснює, які саме слова або фрагменти тексту вплинули на рішення. Це ускладнює діагностику помилок і знижує довіру кінцевих користувачів, особливо якщо система використовується для прийняття важливих рішень (наприклад, модерації контенту).

Залежність від YouTube API – система покладається на офіційний API, який має квоту на кількість запитів на добу. Для великих досліджень або комерційного використання це може стати обмеженням. Крім того, API не надає доступу до коментарів, якщо відео є закритим або автор вимкнув обговорення.

Враховуючи виявлені обмеження, можна окреслити кілька шляхів розвитку системи.

Покращення розпізнавання сарказму – один із підходів – інтегрувати

додатковий контекст, наприклад аналіз самого відео (назва, опис, попередні коментарі) або використовувати ансамбль моделей, де одна відповідає за буквальну тональність, а інша – за виявлення саркастичних маркерів (надмірний капс, повторювані знаки оклику, специфічні емодзі). Також можна збирати розмічені навчальні дані, де сарказм позначено окремо.

Адаптація до конкретних доменів – замість використання універсальної моделі можна виконати донавчання (fine-tuning) на даних конкретного каналу або тематики. Наприклад, для відео про технології лексика буде іншою, ніж для розважальних ток-шоу. Це підвищить точність, але потребує додаткових розмічених даних.

Оптимізація продуктивності – для зменшення часу інференсу можна використати легші версії моделей (наприклад, DistilBERT або XLM-R зі зменшеною кількістю шарів). Інший варіант – виконання обробки на GPU із застосуванням кешування результатів для часто повторюваних коментарів. Також можна розгорнути модель як мікросервіс із використанням спеціалізованих інструментів (ONNX, TensorRT).

Інтеграція додаткових джерел даних – крім тексту коментаря, модель може враховувати метадані: кількість лайків під коментарем (високий рейтинг часто вказує на підтримку думки), тип відео, мову коментаря. Це дозволить точніше визначати тональність у складних випадках.

Пояснення прогнозів – додавання механізмів інтерпретації (наприклад, підсвічування важливих слів у коментарі за допомогою LIME або SHAP) підвищить довіру до системи та спростить відлагодження. Користувач зможе побачити, чому модель віднесла коментар до негативного – через слово «жахливо» або через контекстне поєднання слів.

Розширення мовної підтримки – хоча XLM-R підтримує 100 мов, якість роботи для деяких із них (зокрема для української) може бути нижчою через меншу представленість у навчальному корпусі. Можна виконати додаткове донавчання на великому корпусі україномовних коментарів, щоб підвищити впевненість для позитивного та нейтрального класів.

Аналіз тональності коментарів у соціальних мережах залишається активною галуззю досліджень. У майбутньому можна очікувати появи мультимодальних моделей, які аналізують одночасно текст, відеоряд, аудіо та міміку авторів. Це дозволить точніше розпізнавати сарказм і приховані емоції. З іншого боку, розвиток методів ефективного навчання (few-shot, zero-shot) дасть змогу адаптувати моделі до нових мов та тематик без великих розмічених корпусів. Таким чином, представлена в роботі система є лише першим кроком, який може бути розширений у багатьох напрямках.

ВИСНОВКИ

У ході виконання дипломної роботи було проведено дослідження методів аналізу тональності текстів, розроблено та реалізовано веб-застосунок для аналізу багатомовних коментарів YouTube з використанням трансформерної моделі XLM-R. В процесі виконання роботи було реалізовано наступні етапи:

– проведено аналіз предметної області, визначено особливості текстів соціальних мереж (YouTube-коментарів), які впливають на якість аналізу тональності: неформальність, мультимовність (зокрема українська, англійська), наявність шуму, емодзі, сарказму;

– виконано класифікацію та порівняльний аналіз існуючих методів аналізу тональності (лексиконні, класичне машинне навчання, нейронні мережі) та обґрунтовано вибір трансформерної моделі XLM-R як найбільш ефективної для багатомовних даних;

– розроблено метод попередньої обробки текстів, який включає очищення від шуму, нормалізацію регістру, спеціальну обробку емодзі та субслівну токенізацію, що дозволило зберегти важливу емоційну інформацію та підготувати дані для моделі;

– запропоновано підхід до класифікації на основі попередньо навченої моделі XLM-R (AmaanP314/youtube-xlm-roberta-base-sentiment-multilingual) з використанням тонкого налаштування (fine-tuning), що дозволило адаптувати загальномовну модель до специфіки YouTube-коментарів;

– спроектовано архітектуру веб-застосунку, що складається з серверної частини на Flask (Python) для інтеграції з YouTube API та моделлю ML, та клієнтської частини (HTML/CSS/JS) для візуалізації;

– програмно реалізовано веб-застосунок, який забезпечує повний цикл аналізу: введення URL відео, отримання коментарів через YouTube Data API, їх токенізацію, класифікацію моделлю, відображення статистики (картки, кругова діаграма), таблиці коментарів з фільтрацією та пагінацією, а також

експорт результатів у CSV та JSON;

– проведено експериментальне дослідження на двох різномовних вибірках (англійський ігровий контент та українське ток-шоу) обсягом до 500 коментарів, проаналізовано розподіл класів тональності та середню впевненість моделі.

Вимоги до функціональності системи виконано в повному обсязі. Розроблений веб-застосунок продемонстрував стабільну роботу на різних обсягах даних. Експерименти підтвердили ефективність обраної моделі: для англійських коментарів загальна середня впевненість склала 0,78, для українських – 0,86 (при цьому на українських даних модель очікувано краще розпізнає негативні коментарі). Окремо відзначено стійкість моделі до змішаних текстів та різних мов завдяки архітектурі XLM-R.

Визначено обмеження поточної системи, зокрема проблеми з розпізнаванням сарказму, залежність від обчислювальних ресурсів, чутливість до дисбалансу класів та потреба в якісних розмічених даних. Окреслено напрями подальшого вдосконалення: оптимізація для роботи на CPU, використання легших версій трансформерів, додаткове донавчання на вузькоспеціалізованих корпусах (наприклад, україномовних коментарях), інтеграція методів пояснюваного ШІ (LIME/SHAP) та покращення інтерфейсу для детального відстеження помилок. Використання розробленої системи аналізу тональності дозволить автоматизувати обробку великих масивів коментарів, надаючи користувачам наочну візуалізацію настроїв аудиторії YouTube.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Liu B. Sentiment Analysis and Opinion Mining. – Morgan & Claypool Publishers, 2012. – 167 p.

2. IBM. Sentiment Analysis. – URL: https://www.ibm.com/topics/sentiment-analysis (дата звернення: 08.03.2026).

3. IBM. Natural Language Processing. – URL: https://www.ibm.com/topics/natural-language-processing (дата звернення: 08.03.2026).

4. MonkeyLearn. Sentiment Analysis Guide. – URL: https://monkeylearn.com/sentiment-analysis/ (дата звернення: 08.03.2026).

5. Jurafsky D., Martin J. H. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. – 3rd ed. – 2020. – URL: https://web.stanford.edu/~jurafsky/slp3/ (дата звернення: 08.03.2026).

6. Wikipedia. TF-IDF. – URL: https://en.wikipedia.org/wiki/Tf-idf (дата звернення: 08.03.2026).

7. Wikipedia. Bag-of-Words Model. – URL: https://en.wikipedia.org/wiki/Bag-of-words_model (дата звернення: 08.03.2026).

8. Mikolov T., Chen K., Corrado G., Dean J. Efficient Estimation of Word Representations in Vector Space. – 2013. – URL: https://arxiv.org/abs/1301.3781 (дата звернення: 08.03.2026).

9. Pennington J., Socher R., Manning C. GloVe: Global Vectors for Word Representation. – 2014. – URL: https://aclanthology.org/D14-1162/ (дата звернення: 08.03.2026).

10. Google Developers. Embeddings (Machine Learning Crash Course). – URL: https://developers.google.com/machine-learning/crash-course/embeddings (дата звернення: 08.03.2026).

11. Hochreiter S., Schmidhuber J. Long Short-Term Memory. Neural Computation. – 1997. – URL: https://www.bioinf.jku.at/publications/older/2604.pdf

(дата звернення: 08.03.2026).

12. Kim Y. Convolutional Neural Networks for Sentence Classification. – 2014. – URL: https://arxiv.org/abs/1408.5882 (дата звернення: 08.03.2026).

13. Vaswani A., Shazeer N., Parmar N. та ін. Attention Is All You Need. Advances in Neural Information Processing Systems. – 2017. – Vol. 30. – URL: https://papers.neurips.cc/paper/7181-attention-is-all-you-need.pdf (дата звернення: 08.03.2026).

14. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of NAACL-HLT. – 2019. – P. 4171–4186. – URL: https://aclanthology.org/N19-1423/ (дата звернення: 08.03.2026).

15. Liu Y., Ott M., Goyal N. та ін. RoBERTa: A Robustly Optimized BERT Pretraining Approach. – 2019. – URL: https://arxiv.org/abs/1907.11692 (дата звернення: 08.03.2026).

16. Conneau A., Khandelwal K., Goyal N. та ін. Unsupervised Cross-lingual Representation Learning at Scale. – 2020. – URL: https://arxiv.org/abs/1911.02116 (дата звернення: 08.03.2026).

17. Alammar J. The Illustrated Transformer. – 2018. – URL: https://jalammar.github.io/illustrated-transformer/ (дата звернення: 08.03.2026).

18. Hugging Face. Course: Transformers, NLP and more. – URL: https://huggingface.co/course (дата звернення: 08.03.2026).

19. Hugging Face. Transformers Documentation. – URL: https://huggingface.co/docs/transformers (дата звернення: 08.03.2026).

20. Wikipedia. Precision and Recall. – URL: https://en.wikipedia.org/wiki/Precision_and_recall (дата звернення: 08.03.2026).

21. Wikipedia. Confusion Matrix. – URL: https://en.wikipedia.org/wiki/Confusion_matrix (дата звернення: 08.03.2026).

22. YouTube Data API. – URL: https://developers.google.com/youtube/v3 (дата звернення: 08.03.2026).

23. PyTorch Documentation. – URL: https://pytorch.org/docs/ (дата

звернення: 08.03.2026).

24. Scikit-learn Documentation. – URL: https://scikit-learn.org/stable/ (дата звернення: 08.03.2026).

25. Flask Documentation. – URL: https://flask.palletsprojects.com/ (дата звернення: 08.03.2026).

26. І. Є. Ткаченко. Особливості обробки текстових даних соціальних мереж при аналізі емоційної тотальності/ 30-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті». Зб. матеріалів форуму. Т. 6., Харків: ХНУРЕ. 2026. с. 494-497