Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Центр ННЦЗФН
(повна назва)
Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Розробка інтелектуальної системи підтримки етичної комунікації в онлайн-середовищі на основі аналізу негативних коментарів
(тема)

Виконав:
здобувач четвертого року навчання,
групи ІТШІз-21-1
Данило Плохий
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-професійна
Освітня програма Штучний інтелект
(повна назва освітньої програми)
Керівник ас. Ірина Малєєва
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ _____ Олег ЗОЛОТУХІН
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Центр ННЦЗФН

Кафедра Штучного інтелекту

Рівень вищої освіти перший (бакалаврський)

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

Тип програми освітньо-професійна

Освітня програма Штучний інтелект
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Плохому Данилу Костянтиновичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка інтелектуальної системи підтримки етичної комунікації в онлайн-середовищі на основі аналізу негативних коментарів

затверджена наказом університету від 7 травня 20 25 р. № 80Стз

2. Термін подання студентом роботи до екзаменаційної комісії 24 червня 20 25 р.

3. Вихідні дані до роботи науково-технічні публікації, дані Інтернет-джерел, датасети токсичних коментарів (Jigsaw Toxic Comment Classification, україномовний корпус dardem), попередньо навчені трансформерні мовні моделі (roberta-base, xlm-roberta-large-uk-toxicity, GPT-4o), програмні засоби Python, Transformers, Streamlit, OpenAI API

4. Перелік питань, що потрібно опрацювати в роботі

1) Аналіз предметної галузі та постановка задачі дослідження

2) Методи та моделі для аналізу токсичності та генерації відповідей

3) Проектування та реалізація системи

4) Експериментальна перевірка та оцінка результатів

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	05.05.2025	виконано
2	Аналіз предметної галузі	06.05.2025	виконано
3	Визначення структури системи та технічного	08.05.2025	виконано
	завдання		
4	Підбір моделей і методів аналізу токсичних	10.05.2025	виконано
	повідомлень		
5	Реалізація класифікації токсичності	10.05.2025	виконано
6	Генерація стилізованих відповідей і підбір	20.05.2025	виконано
	промптів		
7	Побудова інтерфейсу та демонстраційної	23.05.2025	виконано
	системи		
8	Експериментальне тестування, візуалізація	26.05.2025	виконано
	результатів		
9	Написання пояснювальної записки	01.06.2025	виконано
10	Перевірка на академічний плагіат	10.06.2025	виконано
11	Нормоконтроль	12.06.2025	виконано
12	Підготовка презентації та доповіді	15.06.2025	виконано
13	Попередній захист	18.06.2025	виконано
14	Рецензування	21.06.2025	виконано
15	Захист перед ЕК	24.06.2025	

Дата видачі завдання 5 травня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ ас. Ірина Малєєва
(підпис) (посада, власне ім’я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 87 с., 41 рис., 1 дод., 29 джерел.

АНАЛІЗ ТЕКСТУ, ГЕНЕРАЦІЯ ВІДПОВІДЕЙ, ЕТИЧНА КОМУНІКАЦІЯ, КЛАСИФІКАЦІЯ КОМЕНТАРІВ, ОБРОБКА ПРИРОДНОЇ МОВИ, ОНЛАЙН-СЕРЕДОВИЩЕ, НЕГАТИВНІ КОМЕНТАРІ, ТОКСИЧНІСТЬ, ШТУЧНИЙ ІНТЕЛЕКТ, GPT-4o, NLP, RoBERTa.

Об’єкт дослідження – процеси текстової комунікації в онлайн-середовищі та їх вплив на етичність взаємодії між користувачами.

Предмет дослідження – методи автоматичного аналізу негативних коментарів та генерації етичних відповідей у цифровому середовищі.

Мета роботи – розробка інтелектуальної системи підтримки етичної комунікації в онлайн-середовищі на основі аналізу негативних коментарів, яка поєднує виявлення токсичності та формування стилістично адаптованих відповідей.

Методи дослідження – трансформерні мовні моделі (RoBERTa, XLM-RoBERTa, GPT-4o), бінарна класифікація, інженерія промптів, генерація тексту через API, поведінкове тестування відповідей.

У роботі реалізовано інтелектуальну систему, що класифікує негативні коментарі за ознакою токсичності та генерує етичні відповіді в трьох стилях: нейтральному, ввічливому та м’якому. Система протестована на прикладах англомовних та україномовних повідомлень з аналізом помилок і можливостей інтеграції в реальні середовища.

ABSTRACT

Bachelor’s thesis contains: 87 pp., 41 fig., 1 ann., 29 references.

ETHICAL COMMUNICATION, GPT-4O, NATURAL LANGUAGE PROCESSING, NEGATIVE COMMENTS, NLP, ONLINE ENVIRONMENT, RESPONSE GENERATION, ROBERTA, TEXT ANALYSIS, TOXICITY, TOXIC COMMENT CLASSIFICATION.

Object of the study: textual communication processes in the online environment and their impact on the ethical quality of user interaction.

Subject of the study: methods of automatic analysis of negative comments and generation of ethical responses in digital communication environments.

Research aim: to develop an intelligent system for supporting ethical communication in the online environment based on the analysis of negative comments, combining toxicity detection and stylized response generation.

Methods: transformer-based language models (RoBERTa, XLM-RoBERTa, GPT-4o), binary classification, prompt engineering, text generation via API, behavioral evaluation of generated responses.

This work presents an intelligent system that classifies negative comments according to their toxicity and generates ethical responses in three styles: neutral, polite, and soft. The system has been tested on examples in both English and Ukrainian, with an analysis of classification errors and integration potential in real-world applications.

ЗМІСТ

ВСТУП

У сучасному світі інформаційні технології є невід'ємною частиною повсякденного життя людей. Активний розвиток онлайн-комунікації відкрив нові можливості для обміну думками, навчання, професійної взаємодії та розваг. Проте водночас із позитивними тенденціями значно зросла і кількість випадків токсичної поведінки у віртуальному просторі.

Токсичність у коментарях та повідомленнях стала серйозною проблемою для онлайн-спільнот, соціальних мереж, форумів і служб підтримки. Ворожі висловлювання, провокації та агресія негативно впливають як на окремих користувачів, так і на загальний рівень культури спілкування в мережі. Це призводить до психологічного дискомфорту, зниження довіри до платформ та посилення соціальної напруги.

У відповідь на виклики токсичності різноманітні компанії та дослідницькі організації розробляють системи моніторингу та модерації контенту. Однак більшість рішень спрямовані переважно на виявлення та блокування агресивних повідомлень, без спроби запропонувати конструктивні підходи до подолання конфліктних ситуацій.

Одним із перспективних напрямів розвитку інтелектуальних систем є створення механізмів, які не лише виявляють негативний контент, але й сприяють підтримці етичної взаємодії між користувачами. Застосування методів обробки природної мови та машинного навчання дозволяє створити рішення, що здатні аналізувати зміст повідомлень і генерувати відповіді відповідно до заданого стилю спілкування.

Особливої уваги вимагає розробка таких систем для україномовного онлайн-середовища, оскільки специфіка мови, культурних норм та особливостей комунікації потребує адаптації існуючих підходів.

У даній роботі розглядається підхід до вирішення проблеми токсичності в онлайн-спілкуванні шляхом створення інтелектуальної системи, яка аналізує негативні коментарі та генерує етичні стилізовані

відповіді. Такий підхід дозволяє не лише зменшити рівень агресії, але й підтримати культуру взаємоповаги в цифровому просторі.

З огляду на важливість етичної взаємодії в цифровому середовищі, особливої актуальності набувають інтелектуальні системи, які здатні не тільки блокувати небажані висловлювання, але й навчати користувачів конструктивного діалогу через відповідну стилізацію відповідей. Таким чином, роль технологій змінюється від контролюючої до виховної, що сприяє формуванню здорової комунікаційної культури.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

У сучасному світі онлайн-комунікація стала однією з основних форм взаємодії між людьми. Завдяки розвитку цифрових технологій, доступності інтернету та мобільних пристроїв, обмін інформацією в режимі реального часу став невід'ємною частиною особистого, професійного та суспільного життя.

Онлайн-комунікація охоплює різноманітні канали – соціальні мережі, форуми, електронну пошту, месенджери, платформи для відеоконференцій. Вона забезпечує доступ до знань, дозволяє швидко вирішувати питання, брати участь у суспільних дискусіях, отримувати підтримку в кризових ситуаціях. За дослідженнями, близько 70% людей у розвинених країнах щодня користуються інтернетом, а відповідно й цифровими каналами для особистої або професійної взаємодії (рисунок 1.1) [1].

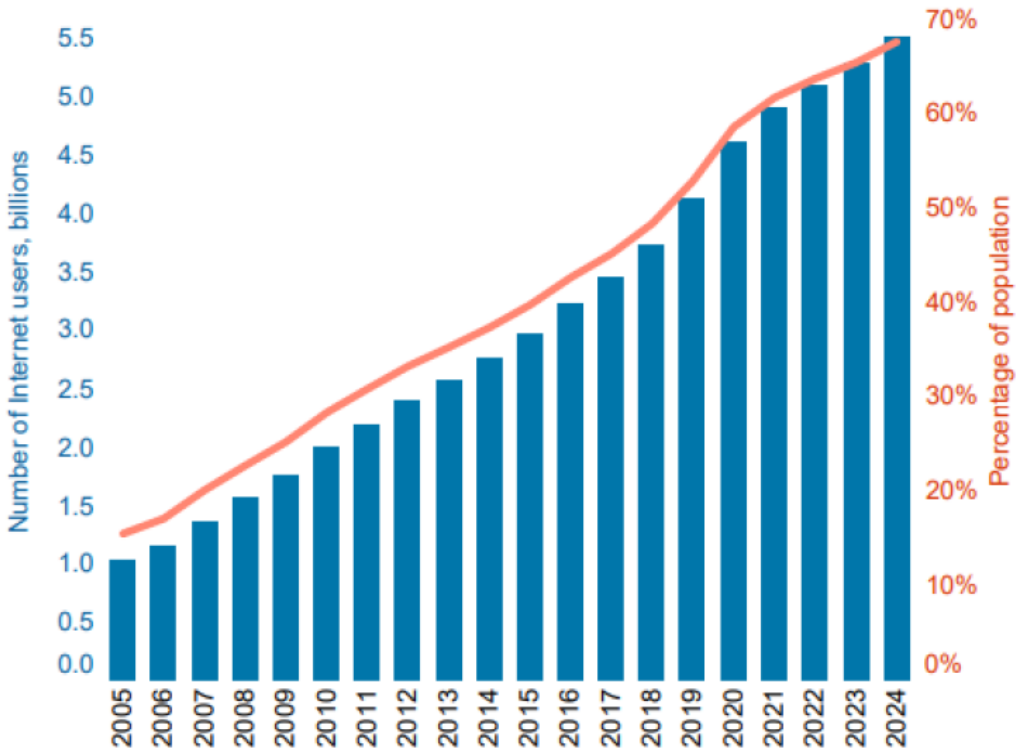


Рисунок 1.1 – Відсоток користувачів інтернету

Ключовими перевагами онлайн-комунікації є її миттєвість, відсутність географічних обмежень, можливість спілкуватися з великою кількістю людей одночасно, а також зручність збереження та обробки інформації. У сфері освіти та бізнесу це дозволило перейти до нових моделей співпраці, зокрема дистанційного навчання, роботи в міжнародних командах, участі в глобальних проєктах.

Проте поряд із беззаперечними перевагами онлайн-комунікація має і певні ризики. Зокрема, втрата невербальних сигналів (міміки, інтонації) може призводити до непорозумінь; анонімність сприяє зниженню відповідальності за висловлювання; легкість поширення інформації інколи створює умови для розповсюдження неправдивих або агресивних повідомлень [2].

Таким чином, онлайн-комунікація відіграє критично важливу роль у сучасному суспільстві, але її стрімкий розвиток одночасно загострює питання культури спілкування, етичної відповідальності та необхідності захисту користувачів від негативного інформаційного впливу. Ці аспекти стають особливо актуальними у зв'язку з масштабністю використання цифрових платформ в Україні та світі.

1.1 Поняття токсичності в онлайн-комунікації

У сучасному цифровому просторі термін «токсичність» широко застосовується для позначення агресивної, ворожої або принизливої поведінки користувачів в онлайн-середовищі. Токсичність у текстових коментарях проявляється у формі образ, принижень, провокацій, маніпуляцій, нецензурної лексики, а також у використанні агресивного або саркастичного тону для навмисного викликання негативних емоцій у співрозмовника.

Згідно з дослідженнями Vidgen та Harris (2020), токсичністю вважаються повідомлення, які провокують агресію, обурення або бажання припинити участь у дискусії [3].

Онлайн-комунікація, на відміну від особистого спілкування, позбавлена важливих невербальних сигналів – таких як міміка, жести, інтонація. Це значно ускладнює процес інтерпретації емоційного забарвлення повідомлень і часто призводить до нерозуміння, а отже, до конфліктів. В анонімному середовищі інтернету користувачі почуваються менш обмеженими у своїх висловлюваннях, що додатково сприяє поширенню токсичних практик спілкування.

Поняття токсичності охоплює широкий спектр поведінки, який може варіюватися від тонкого сарказму і пасивної агресії до відкритої образи чи залякування. За класифікацією дослідників Google Jigsaw (розробників Perspective API), токсичними вважаються ті висловлювання, які викликають у читача відчуття агресії, образи або бажання припинити дискусію [4]. Також у контексті дослідження онлайн-спільнот виділяють поняття «шкідливої комунікації» (harmful communication), що включає дезінформацію, маніпуляцію, булінг і мову ворожнечі.

Окрім прямих образ, токсичність може проявлятися у формі провокативних запитань, упереджених тверджень, маніпулятивного сарказму або прихованих натяків на приниження. Особливо небезпечною є токсичність у спільнотах, орієнтованих на вразливі категорії користувачів – підлітків, осіб із психологічними травмами або хронічними захворюваннями.

Поширення токсичних моделей поведінки в мережі має численні негативні наслідки. Серед них – погіршення психологічного стану користувачів, зниження довіри до онлайн-спільнот, формування культури агресивної взаємодії. За даними досліджень Американської психологічної асоціації, регулярне зіткнення з токсичністю в інтернеті може призводити

до підвищеного рівня тривожності, депресії, ізоляції та втрати впевненості у собі.

Проблема токсичності в онлайн-комунікації вимагає не тільки її загального розуміння, а й чіткого класифікування конкретних проявів. Це дозволяє розробляти більш ефективні стратегії виявлення та нейтралізації агресивних висловлювань. Відповідно до сучасних досліджень, можна виділити кілька основних типів токсичних повідомлень у цифровому середовищі:

– відкрита образа: найочевидніший тип токсичності, що виражається у прямих агресивних висловлюваннях, спрямованих на приниження, ображання або дискредитацію іншої особи. Часто супроводжується використанням нецензурної лексики;

– пасивна агресія: більш прихована форма токсичності, коли негативне ставлення проявляється через сарказм, натяки, недоброзичливі коментарі, замасковані під «жарти». Такий тип складніше виявити автоматичними засобами через тонку емоційну забарвленість тексту;

– провокаційні повідомлення (тролінг): навмисне публікування провокативних висловлювань або інформації з метою викликати гнів, сварку чи агресивну реакцію у співрозмовників. Тролінг часто має на меті емоційне розгойдування спільнот;

– мова ворожнечі (хейт-спіч): коментарі, спрямовані на розпалювання ненависті за ознакою раси, національності, статі, релігії або інших ознак. Цей тип токсичності регулюється навіть на законодавчому рівні у багатьох країнах;

– маніпулятивна токсичність: використання тактики маніпуляції з метою змусити співрозмовника почуватися винним, неповноцінним або змінити свою позицію через психологічний тиск. Такий вид токсичності часто залишається непоміченим без глибокого аналізу тексту;

– інформаційна агресія: навмисне поширення неправдивої, викривленої або провокативної інформації з метою дестабілізації дискусій або дезорієнтації користувачів.

Кожен з наведених типів токсичних повідомлень має власні особливості, однак усі вони негативно впливають на емоційний клімат спільноти та якість онлайн-взаємодії. Правильна класифікація допомагає не лише виявляти токсичні повідомлення, а й обирати найбільш адекватні способи реагування на них – від попередження і модерації до формування відповідей у відповідному стилі.

Розуміння різновидів токсичної поведінки в онлайн-середовищі дозволяє оцінити масштаби її негативного впливу на окремих користувачів, спільноти та суспільство загалом. Токсичність не лише ускладнює спілкування між людьми, а й має реальні психологічні, соціальні та культурні наслідки.

На індивідуальному рівні зіткнення з агресивними або образливими повідомленнями може спричинити широкий спектр емоційних реакцій: від короткочасного стресу до розвитку стійких тривожних станів, депресії чи навіть соціальної ізоляції. Особливо вразливими до негативного впливу є підлітки, особи із низьким рівнем соціальної підтримки та користувачі, які перебувають у кризових психологічних станах.

Психологічні дослідження показують, що регулярний контакт із токсичними повідомленнями здатен змінювати уявлення людини про себе та про світ загалом. Це призводить до зниження самооцінки, підвищення рівня недовіри, розвитку почуття ворожості навіть у нейтральних взаємодіях. Деякі користувачі внаслідок токсичних атак ухвалюють рішення обмежити свою активність у соціальних мережах або повністю їх залишити.

На рівні спільнот токсична поведінка підриває фундаментальні принципи здорової комунікації: повагу, відкритість до діалогу, готовність сприймати різні думки. У спільнотах, де агресивні висловлювання

залишаються безкарними, з часом формується атмосфера ворожості, яка відштовхує нових учасників і провокує подальшу ескалацію конфліктів.

Для суспільства в цілому поширення токсичності в онлайн-просторі має серйозні наслідки. Масове розповсюдження мови ворожнечі сприяє радикалізації поглядів, зростанню соціальної напруженості, поглибленню поділів за різними ознаками (політичними, етнічними, релігійними) [5]. Крім того, токсичне середовище в інтернеті може впливати на реальні події, зокрема підбурювати до актів насильства або поглиблювати конфлікти у суспільстві.

Інформаційні платформи, що не вживають ефективних заходів проти токсичності, ризикують втратити довіру користувачів, а відповідно – й конкурентоспроможність на ринку. У відповідь на це багато міжнародних корпорацій уже впроваджують політики «нульової толерантності» до агресивної поведінки та розробляють інструменти для її автоматичного виявлення і блокування.

Різні платформи по-різному визначають і борються з токсичністю. Деякі сервіси (наприклад, YouTube, Twitter) впроваджують автоматичне виявлення образливих коментарів, деякі (Facebook, Instagram) акцентують на модерації вмісту за участі людини. Однак проблема залишається актуальною через складність точного визначення межі між допустимою критикою та реальною токсичністю.

Усвідомлення масштабів і наслідків токсичної поведінки в онлайн-середовищі неможливе без аналізу чинників, що сприяють її поширенню. Існує низка особливостей цифрового простору, які створюють сприятливе середовище для виникнення та закріплення токсичних практик у спілкуванні.

Однією з основних причин є анонімність користувачів. Інтернет забезпечує можливість залишатися невідомим або створювати фальшиві профілі, що суттєво знижує рівень особистої відповідальності за власні

висловлювання. За відсутності страху перед наслідками люди часто дозволяють собі те, чого б не сказали у реальному житті.

Відсутність негайної соціальної реакції є ще одним важливим чинником. У традиційній комунікації негативні висловлювання супроводжуються миттєвою емоційною відповіддю співрозмовника, яка стримує подальшу агресію. У віртуальному просторі така реакція або відсутня, або значно відтермінована, що послаблює самоконтроль.

Ефект «деіндивідуалізації», описаний у соціальній психології, також активно діє в мережевому середовищі. У великих анонімних спільнотах особистість окремого учасника «розчиняється» у натовпі, внаслідок чого люди схильні діяти імпульсивніше, агресивніше і менш відповідально [6].

Свою роль відіграють і алгоритмічні механізми цифрових платформ. Системи рекомендацій часто сприяють поширенню емоційно забарвленого контенту, зокрема й агресивного, оскільки саме такий контент забезпечує більшу залученість користувачів (більше коментарів, репостів, лайків).

Додатково, соціальні, політичні та економічні чинники можуть підсилювати рівень загальної напруги у суспільстві, що відображається в агресивнішій поведінці користувачів в інтернеті.

1.2 Огляд існуючих рішень для виявлення та обробки токсичних повідомлень

Різні компанії та дослідницькі організації впроваджують рішення, що базуються на різноманітних технологіях: від простих лексичних фільтрів і правил до складних моделей машинного навчання та глибокого навчання. Застосування методів обробки природної мови (NLP) та нейронних мереж дозволяє суттєво підвищити точність виявлення токсичних висловлювань, проте одночасно створює нові виклики – зокрема, адаптацію моделей до контексту та різноманітних мов.

Perspective API – це платформа, розроблена лабораторією Jigsaw (підрозділ Google), яка надає інструменти для автоматичного аналізу токсичності в текстах. Сервіс використовує моделі машинного навчання для оцінки ймовірності того, що певний коментар буде сприйнятий як образливий, провокаційний або шкідливий [7].

Основною метою Perspective API є допомога модераторам в управлінні великими обсягами користувацького контенту, шляхом автоматичного виявлення проблемних висловлювань. API оцінює коментар за різними показниками: токсичність, загроза, образливість, агресія, мова ненависті тощо (рисунок 1.2). Результати повертаються у вигляді шкали від 0 до 1, що дозволяє інтегрувати систему у модераційні інструменти платформ.

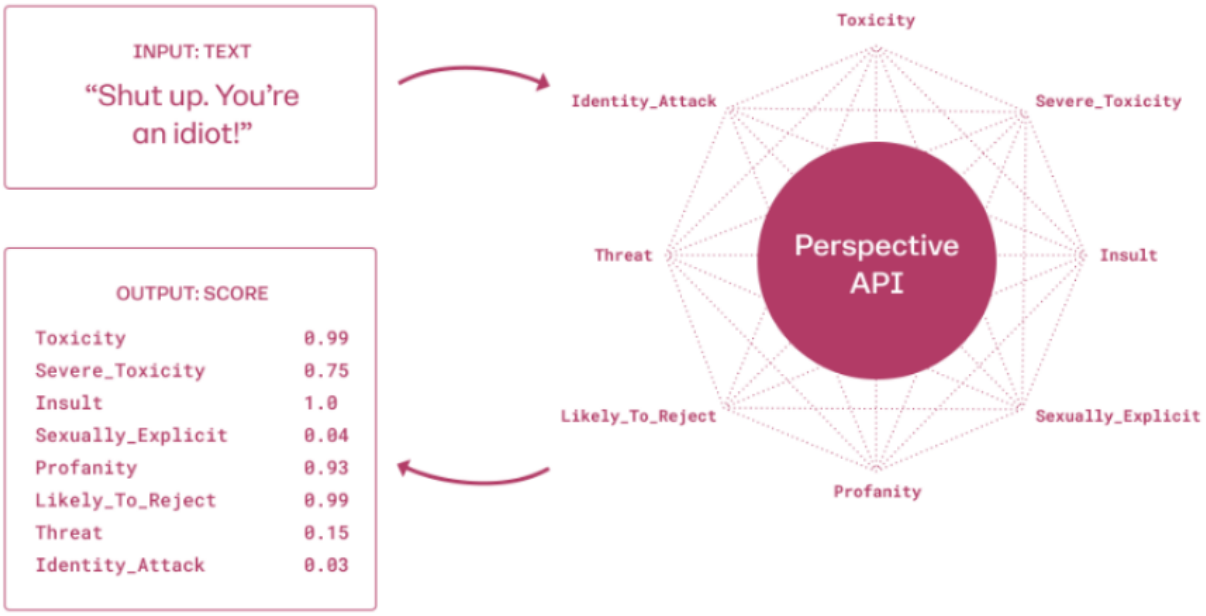


Рисунок 1.2 – Схема роботи Perspective API

Технологічно Perspective API базується на моделях глибокого навчання, зокрема на конволюційних і рекурентних нейронних мережах, натренованих на великих відкритих корпусах коментарів із позначеннями токсичності, таких як датасет Wikipedia Detox.

Серед переваг Perspective API варто відзначити високу точність аналізу, відкритість API для розробників та постійне оновлення моделей. Недоліком є те, що сервіс переважно орієнтований на англомовний контент; підтримка інших мов, включно з українською, є обмеженою і потребує додаткової адаптації або тренування моделей на специфічних локалізованих даних.

DeepMoji – це модель, створена дослідниками MIT для розпізнавання емоційного забарвлення в тексті за допомогою емодзі як навчальних міток. Хоча ця система не була розроблена спеціально для детекції токсичності, вона стала важливим інструментом у суміжних дослідженнях, зокрема в задачах визначення агресії, сарказму, образливого тону та пасивної агресії.

Ідея DeepMoji полягає в тому, що за допомогою великого корпусу Twitter-повідомлень (1,2 млрд твітів), які містять емодзі, можна навчити модель розпізнавати тон повідомлення. У процесі тренування емодзі використовувалися як «сурогатні емоційні мітки», що дозволяло моделі вивчати емоційну структуру тексту без ручної розмітки (рисунок 1.3).

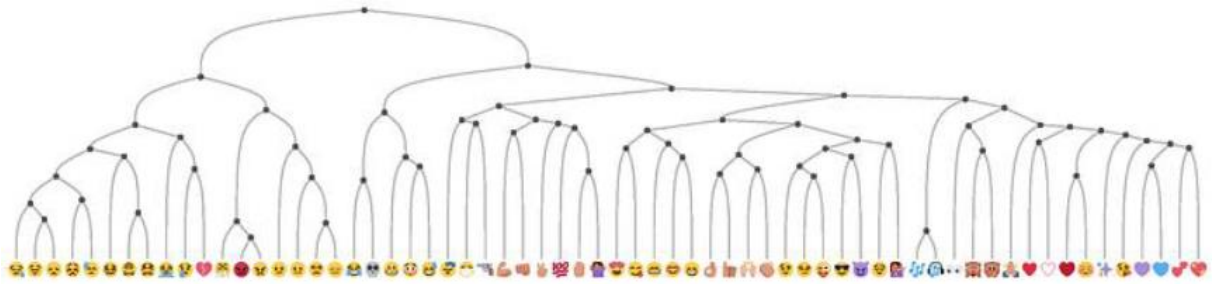


Рисунок 1.3 – Кластеризація емодзі в DeepMoji

Архітектура DeepMoji базується на двох бінаправлених LSTM-рівнях з механізмом attention, що дає змогу ефективно враховувати контекст у текстах [8]. Після попереднього навчання модель була донавчена для задач емоційної класифікації, включно з hate speech, кібербулінгом та токсичною лексикою.

Основна перевага DeepMoji – здатність вловлювати емоційні нюанси тексту, включно з тонкою іронією чи сарказмом, які часто складно класифікувати традиційними підходами. Завдяки своїй гнучкості модель часто використовують як попередній етап для емоційної нормалізації текстів або як частину складніших систем токсичності.

Однак використання DeepMoji має певні обмеження. Зокрема, модель навчена на специфічному домені коротких повідомлень з Twitter, що може впливати на її ефективність у довших або формальніших текстах. Також слід враховувати, що використання емодзі як сурогатних емоційних міток не завжди точно відображає реальні інтенції автора, особливо в контексті токсичності, де емоційне забарвлення може бути прихованим або двозначним.

Moderation API – це рішення від OpenAI, розроблене для автоматичного виявлення небажаного або небезпечного контенту в текстах. Воно є частиною захисної інфраструктури великих мовних моделей (зокрема GPT), однак може також використовуватись як окремий інструмент для модерації користувацьких повідомлень.

API класифікує текст за низкою категорій, серед яких: hate, harassment, violence, self-harm, sexual, self-harm/intent та інші (рисунок 1.4). Система працює на базі спеціалізованих fine-tuned моделей, які навчені розпізнавати порушення політик OpenAI. Модель надає бінарну оцінку (порушує чи ні) для кожної категорії, а також ймовірність для кожного з класів, що дозволяє гнучко застосовувати пороги прийняття рішень.

Moderation API орієнтований насамперед на захист від зловживань, тому особливу увагу приділяє темам чутливого контенту: агресія, ворожість, сексуалізовані повідомлення, загрози тощо. Його основними сценаріями використання є фільтрація вхідних запитів до мовних моделей, а також попередня модерація тексту в чатах, форумах, ігрових сервісах [9].

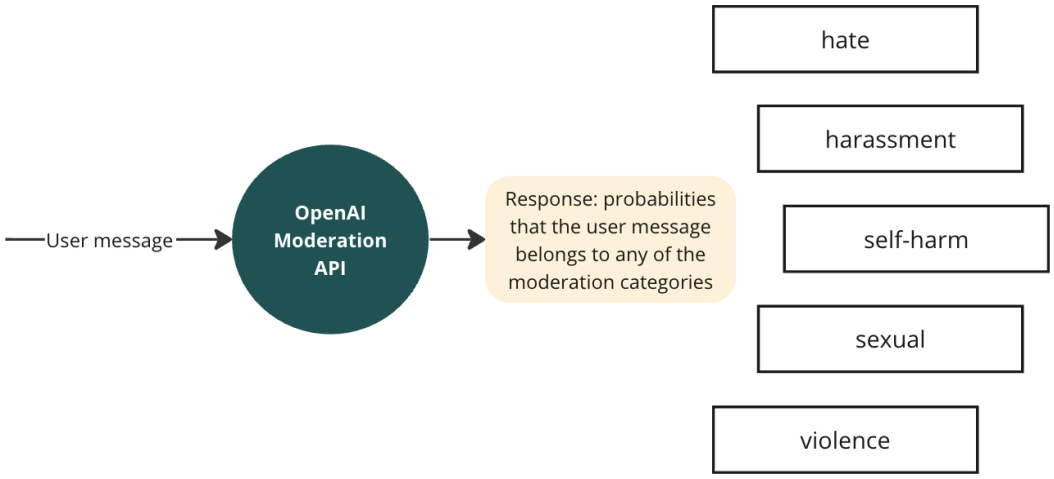


Рисунок 1.4 – Робота OpenAI Moderation API

Перевагою API є його висока інтегрованість у сучасні AI-системи та регулярне оновлення з боку OpenAI. API добре документований, швидкий у роботі й не потребує навчання з нуля – достатньо просто викликати модель і проаналізувати відповідь.

Серед обмежень варто відзначити те, що модель є закритою – немає доступу до деталей навчального набору або архітектури, а сам API не дозволяє модифікувати або адаптувати модель під конкретні потреби. Також деякі категорії класифікації можуть бути надто широкими або контекстуально неоднозначними, що іноді призводить до хибнопозитивних спрацювань.

ToxiGuard [10] – це експериментальна система, розроблена з метою виявлення та усунення токсичних висловлювань із текстів за допомогою методів обробки природної мови (NLP) та класичних алгоритмів машинного навчання. Основна ідея ToxiGuard полягає у виявленні токсичних елементів тексту з подальшим їх очищенням без порушення смислу повідомлення.

Особливістю підходу є саме орієнтація на «очищення» тексту, а не просто на виявлення. Це дозволяє інтегрувати ToxiGuard у середовища, де важливо зберігати повідомлення (наприклад, освітні чати, ігрові форуми), але при цьому уникати поширення токсичних висловлювань.

Переваги:

– простота реалізації та швидкість роботи;

– низькі обчислювальні витрати – система підходить для роботи в реальному часі;

– прозорість – легко інтерпретувати, які слова були класифіковані як токсичні.

Недоліки:

– модель не враховує контекст – токсичність класифікується лише на рівні окремих слів;

– наявний ризик «надмірного очищення», коли нейтральні або саркастичні фрази помилково видаляються;

– обмежена ефективність щодо складних граматичних конструкцій або мовних ідіом.

Останні дослідження у сфері боротьби з онлайн-токсичністю дедалі частіше звертаються до потенціалу великих мовних моделей (LLM), таких як GPT-3 або FLAN-T5. У статті [11] автори досліджують, наскільки ефективними можуть бути ці моделі при використанні спеціально сформульованих інструкцій (промптів), без потреби у традиційному навчанні на токсичних датасетах.

У цьому підході мовну модель просять виконати конкретні завдання: визначити, чи є текст токсичним, пояснити чому саме, або навіть перефразувати його у нейтральному вигляді. Такі задачі модель виконує, маючи лише промпт – текстову інструкцію, яка формує бажану поведінку. Експерименти показали, що навіть без додаткового навчання моделі здатні точно класифікувати токсичний вміст, особливо у варіантах із кількома прикладами (few-shot) або з поясненням логіки (chain-of-thought). Результати були порівняні з більш традиційними моделями, такими як Toxic-BERT, і виявилося, що промптовані LLM можуть досягати подібного або навіть вищого рівня точності.

Однією з важливих переваг цього підходу є його гнучкість – достатньо змінити промпт, аби адаптувати модель під новий контекст або тип токсичності. Окрім цього, великі моделі здатні не лише класифікувати повідомлення, а й переформулювати їх, зберігаючи зміст, але пом’якшуючи агресивний тон. Це відкриває можливості для створення не просто систем виявлення токсичності, а повноцінних етичних асистентів.

Разом з тим, варто зважати на низку обмежень: високі обчислювальні витрати, необхідність API-доступу до LLM, а також значна залежність результатів від якості сформульованої інструкції.

У роботі [12] автор пропонує підхід до виявлення токсичних коментарів з одночасним зменшенням ненавмисної упередженості моделей, пов'язаної з ідентичнісними ознаками, такими як раса, гендер, релігія тощо. Для цього використовується донавчена модель BERT (рисунок 1.5) з ваговими втратами, що дозволяє враховувати дисбаланс у даних та зменшувати упередженість.

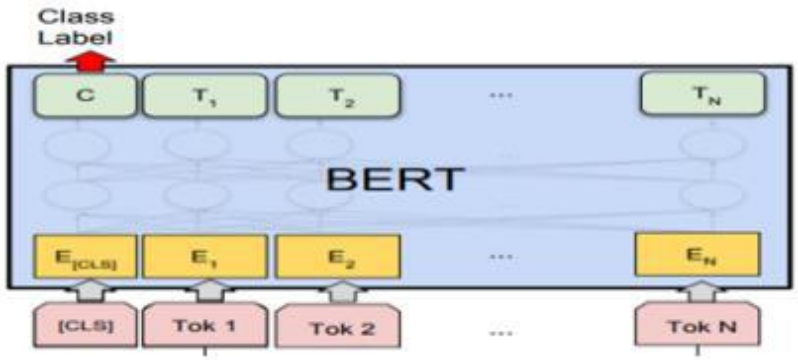


Рисунок 1.5 – BERT у задачі виявлення токсичних повідомлень

У дослідженні порівнюється ефективність донавченої моделі BERT з традиційною логістичною регресією. Результати показують, що модель BERT досягає точності 89% у виявленні токсичних коментарів, тоді як логістична регресія – лише 57.1%. Код проєкту доступний на GitHub, що дозволяє відтворити результати та адаптувати модель до власних потреб.

Однією з головних переваг запропонованого підходу є використання моделі BERT, яка демонструє високу точність при класифікації токсичних повідомлень, навіть у випадках, коли мова йде про приховану агресію або лексичну варіативність. Використання вагових коефіцієнтів у функції втрат дозволяє моделі зменшувати ризик дискримінації по відношенню до певних груп, що є надзвичайно важливим для практичного застосування систем у публічних середовищах.

Втім, незважаючи на очевидні переваги, підхід має і певні обмеження. По-перше, модель BERT потребує значних обчислювальних ресурсів для донавчання, що може ускладнювати її використання у невеликих проєктах або в реальному часі. По-друге, хоча автор дослідження пропонує механізм зниження упередженості, питання пояснюваності рішень моделі залишається відкритим – система не забезпечує прозорого механізму аргументації, чому саме те чи інше повідомлення було класифіковане як токсичне. Це може стати перешкодою для впровадження у чутливих сферах, де важливо обґрунтовувати рішення автоматизованих систем.

Модель DeepHate, запропонована в дослідженні Rui Cao та ін. [13], поєднує різні аспекти текстового представлення, включаючи векторні представлення слів, емоційні тональності та тематичну інформацію, для покращення виявлення мови ворожнечі. Цей підхід дозволяє моделі враховувати не лише лексичні особливості, але й контекстуальні та семантичні аспекти повідомлень.

Переваги:

– інтеграція різних рівнів текстової інформації для більш точного виявлення;

– покращення результатів на кількох реальних датасетах.

Недоліки:

– підвищена складність моделі може ускладнити її впровадження в реальних умовах.

У роботі [14] автори пропонують інноваційну архітектуру AngryBERT, яка не обмежується класифікацією висловлювання як токсичного, а одночасно визначає, на кого саме спрямовано це висловлювання (наприклад, етнічна чи соціальна група), а також емоційний тон, з яким воно подано.

Архітектурно модель поєднує BERT із додатковими двонапрямленими LSTM-рівнями, які обробляють текст окремо для кожного завдання (рисунок 1.6). На етапі об’єднання інформації використовується механізм «gate fusion», що дозволяє інтегрувати представлення з усіх задач для досягнення узгодженого прогнозу. Таким чином, модель навчається не лише розпізнавати прямі прояви токсичності, а й вловлювати тональність і цілеспрямованість повідомлення – важливі аспекти для виявлення менш очевидних форм мови ворожнечі.

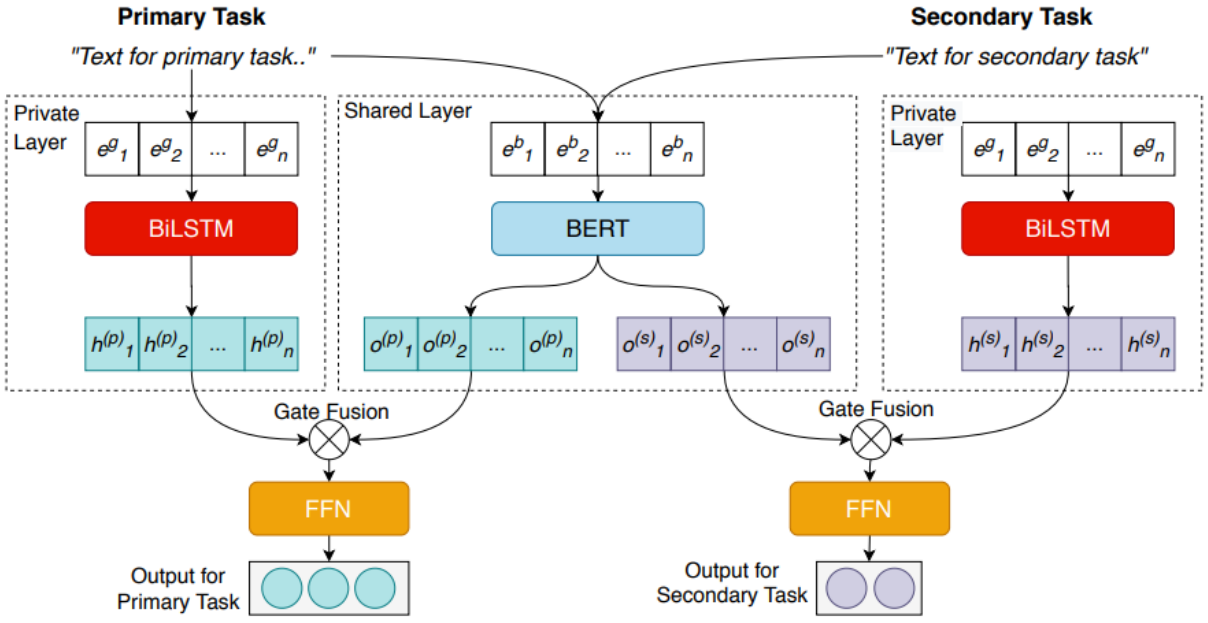


Рисунок 1.6 – Архітектура багатозадачної моделі AngryBERT

Результати експериментів засвідчили, що спільне навчання задачі класифікації разом із задачами емоційного аналізу та цільової ідентифікації дає суттєве покращення у точності. Особливо це помітно у випадках

латентної токсичності, де важливий контекст та соціальний підтекст. Модель дозволяє мінімізувати хибнопозитивні спрацювання за рахунок ширшого розуміння структури повідомлення.

Недоліком підходу є його складність: для ефективного навчання AngryBERT вимагає якісно анотованих даних одночасно за кількома критеріями, а також суттєвих обчислювальних ресурсів. Впровадження такої системи у реальне середовище потребує високого рівня технічної підготовки та доступу до великих корпусів анотацій.

У роботі [15] автори пропонують підхід, який поєднує контрольоване та самоконтрольоване контрастивне навчання з метою побудови більш інформативного простору ознак. Ключовим елементом цієї архітектури є формування пар – позитивних і негативних прикладів, – що дозволяє моделі краще виявляти подібність або відмінність між вхідними повідомленнями.

Модель починає з BERT-ембеддингів, які доповнюються через процедуру data augmentation – парафразування, підстановки чи інші трансформації тексту. Далі формуються пари прикладів, які потрапляють до модуля подвійного контрастивного навчання: самонавчання працює на узагальнених структурах даних, а контрольоване – з урахуванням наявних міток. Це дозволяє навчитись не лише класифікувати токсичний контент, а й будувати стійке уявлення про подібність між прикладами в просторах ознак. На завершальному етапі модель інтегрує контрастивні втрати з класичною функцією втрат (focal loss), спрямованою на боротьбу з дисбалансом класів (рисунок 1.7).

Основною перевагою такого підходу є здатність розпізнавати неочевидні прояви мови ворожнечі, зокрема замасковані під іронію чи сарказм висловлювання. Завдяки контрастивному навчанню модель формує глибше розуміння структури висловлювання, ніж при звичайній класифікації. Крім того, подвійна схема навчання дозволяє зменшити потребу у великих анотованих датасетах, оскільки частину роботи виконує самонавчальна гілка.

Недоліком є складність реалізації – як у технічному, так і в обчислювальному сенсі. Побудова якісних пар даних та керування кількома функціями втрат вимагає детального налаштування. Окрім того, підвищення точності досягається ціною значного ускладнення архітектури та довшого часу тренування, що може обмежити застосування у задачах з обмеженими ресурсами.

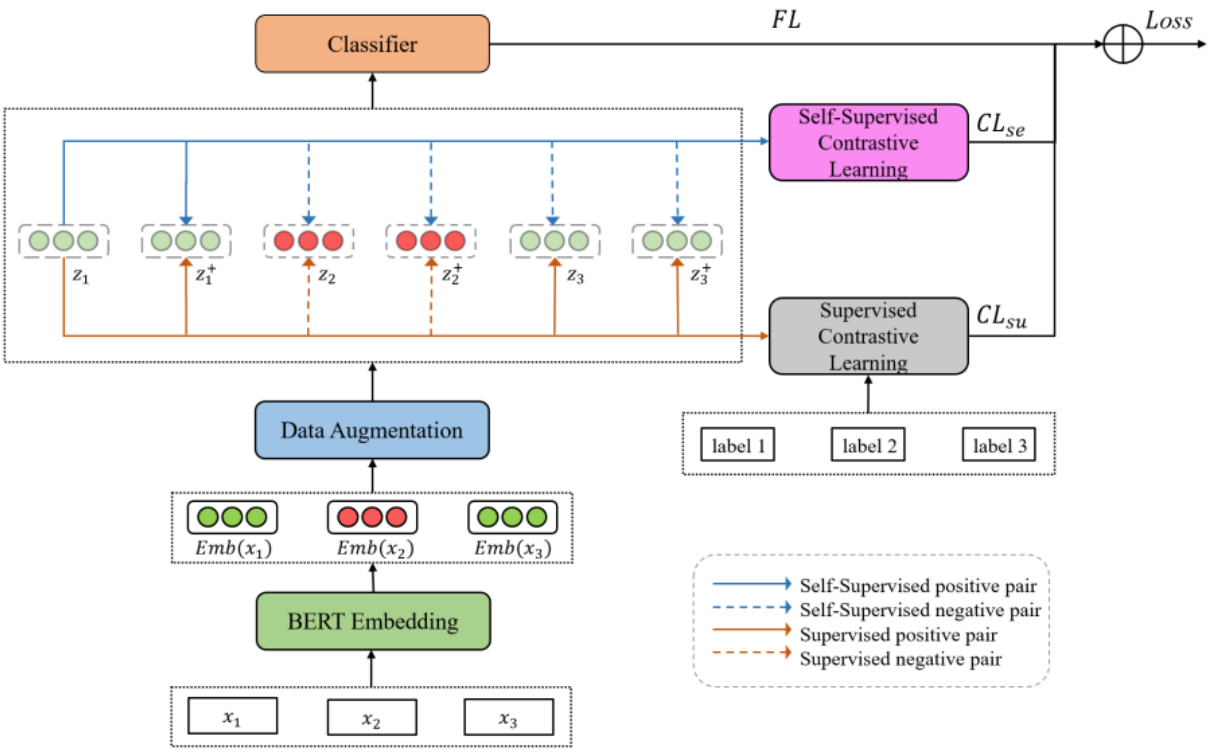


Рисунок 1.7 – Архітектура моделі для виявлення мови ворожнечі з подвійним контрастивним навчанням

Розглянуті рішення охоплюють широкий спектр підходів до виявлення токсичного контенту – від класичних моделей машинного навчання до сучасних глибинних нейронних мереж із багатозадачним або контрастивним навчанням. Таке різноманіття демонструє, наскільки багатовимірною є проблема токсичності: вона вимагає врахування лексики, емоційного тону, контексту та навіть соціальних підтекстів.

1.3 Постановка задачі дослідження

З огляду на вищезазначене, проблема автоматичного виявлення токсичності є однією з ключових задач сучасної обробки природної мови. Система, здатна аналізувати повідомлення та класифікувати їх за рівнем токсичності, дозволяє не лише зменшити негативний вплив агресивного контенту, а й підтримати культуру цивілізованого спілкування у цифровому середовищі.

Метою цієї роботи є створення інтелектуальної системи, що поєднує автоматичне виявлення токсичних повідомлень та генерацію етичних відповідей у стилістично адаптованій формі.

Для досягнення поставленої мети необхідно розв’язати такі задачі:

– проаналізувати теоретичні аспекти токсичності як явища у цифровому спілкуванні, а також оглянути сучасні системи її автоматичного виявлення;

– визначити вимоги до інтелектуальної системи підтримки етичної комунікації, сформулювати обмеження та типові сценарії застосування;

– обґрунтувати вибір мовних моделей для класифікації токсичності англійською та українською мовами та для генерації відповідей;

– реалізувати класифікатор токсичних коментарів на основі попередньо навчених трансформерних моделей (RoBERTa, XLM-RoBERTa);

– реалізувати модуль генерації відповідей у трьох стилях (нейтральному, ввічливому, м’якому) із використанням інструкційного підходу (prompt engineering) на основі GPT-4o;

– інтегрувати обидва модулі в єдину систему з текстовим і веб-інтерфейсами;

– провести тестування системи на прикладах англомовних та україномовних коментарів, здійснивши поведінковий аналіз та аналіз помилок генерації й класифікації.

2 МЕТОДИ ТА МОДЕЛІ ДЛЯ АНАЛІЗУ ТОКСИЧНОСТІ ТА ГЕНЕРАЦІЇ ВІДПОВІДЕЙ

Розробка систем, здатних розпізнавати токсичність і формувати адекватну реакцію, ґрунтується на поєднанні різнотипних підходів до обробки природної мови. З одного боку – моделі класифікації, що виявляють ознаки агресії, з іншого – генеративні архітектури, які забезпечують побудову змістовної відповіді. Кожна з цих компонент базується на окремій методологічній традиції: від простих евристик до трансформерних моделей останнього покоління. Перш ніж перейти до опису конкретної реалізації, важливо розібратися, які саме алгоритми і архітектури використовуються для виявлення токсичності та автоматичної генерації тексту, які з них ефективні на практиці і чим вони відрізняються між собою.

2.1 Методи виявлення токсичності

Визначення токсичності в тексті – одне з базових завдань у побудові систем мовної модерації. Залежно від складності контенту, обсягу даних і вимог до точності, для цієї задачі застосовуються різні підходи: від простих правил до повноцінних трансформерних моделей. Кожен з них має свої технічні переваги й обмеження, що напряму впливають на придатність до конкретного середовища чи типу комунікації.

2.1.1 Rule-based підходи

Rule-based підходи були одними з перших, що застосовувалися для автоматичного виявлення неприйнятного контенту в онлайн-середовищі. Їхня суть полягає у використанні заздалегідь сформованих правил – найчастіше це словники заборонених слів, шаблони фраз або регулярні

вирази, які сигналізують про наявність агресивного чи образливого висловлювання. Такі системи не потребують навчання, працюють швидко й прозоро: результат залежить напряму від того, як побудований набір правил.

У простих випадках rule-based підхід може виявитися ефективним – наприклад, при виявленні прямої лексичної агресії, нецензурної лексики або закликів до насильства. Проте саме така прямолінійність і є головним обмеженням: система не враховує контекст, не здатна розрізнити іронію, метафору або неоднозначне значення. Наприклад, фраза з формально нейтральною лексикою може бути токсичною в конкретній ситуації, тоді як вживання «забороненого» слова – цілком прийнятним у саркастичному чи мемному контексті.

Ще однією проблемою rule-based рішень є те, що користувачі швидко адаптуються до їхніх обмежень. Токсичний зміст часто навмисно маскується шляхом викривлення слів, вставляння символів або використання синонімів, що не охоплені в словнику. Тому ефективність таких систем швидко знижується без постійного оновлення правил, що потребує залучення людини і не масштабується для великих платформ.

Для підвищення точності rule-based підходів у межах GermEval 2021 було запропоновано комбінувати автоматичне генерування шаблонів токсичних висловлювань із ручним коригуванням, що дозволило створити високоточні списки ключових слів [16]. Зокрема, було виявлено, що використання у правилах таких маркерів, як повторювані слова у верхньому регістрі або специфічна лексика (наприклад, образливі іменники), може забезпечити до 91% точності за окремими правилами.

У ході дослідження для побудови таких правил використовувалися як уніграми, так і біграми, отримані з корпусу німецькомовних коментарів, попередньо анотованих як токсичні. Однак навіть найкращі словникові правила виявили обмежений охват – їхній recall залишався низьким, що вказує на необхідність гібридних рішень. Одним із таких рішень стало об’єднання rule-based та BERT-моделей у ансамблеві системи, де перші

виконують роль «першого фільтра», а другі – забезпечують глибший контекстуальний аналіз.

Попри ці обмеження, rule-based підходи залишаються актуальними як перший фільтр – дешевий, швидкий і простий у реалізації. У поєднанні з більш складними моделями вони можуть виступати як допоміжний інструмент, знижуючи навантаження на основну систему або дозволяючи оперативно блокувати очевидно неприйнятні повідомлення ще до запуску повного аналізу.

Також важливо, що такі системи легко піддаються налаштуванню кінцевими користувачами, адже кожне правило є прозорим і контрольованим, що дозволяє швидко виправити хибні спрацьовування або додати нові шаблони за потреби.

2.1.2 Класичні ML-моделі

Класичні моделі машинного навчання стали логічним наступним етапом після rule-based систем у задачах фільтрації токсичного контенту, оскільки дозволяють автоматично виявляти закономірності в тексті без необхідності вручну формулювати правила. Замість фіксованого набору ключових слів, вони використовують статистичні ознаки, які будуються на основі розподілу слів у корпусі. Це робить моделі більш гнучкими, адаптивними до нових формулювань і менш залежними від повноти словника. В основі таких підходів лежить ідея перетворення тексту на числові вектори, які зберігають найважливішу інформацію про структуру та вміст коментаря.

Одним з найбільш поширених способів такого подання є TF-IDF (Term Frequency-Inverse Document Frequency), його основна ідея полягає в тому, щоб поєднати два аспекти інформації про слово: наскільки часто воно трапляється у конкретному документі (TF) та наскільки рідкісним воно є в усій колекції документів (IDF). Якщо слово зустрічається

часто в одному документі, але рідко в інших – воно отримує високу вагу, оскільки вважається інформативним. Натомість слова, які часто трапляються в усіх документах (наприклад, службові частини мови або загальновживані слова), мають низький IDF, що знижує їхній вплив на класифікацію. На рисунку 2.1 наведена формула розрахунку ваги терміна за методом TF-IDF [17].

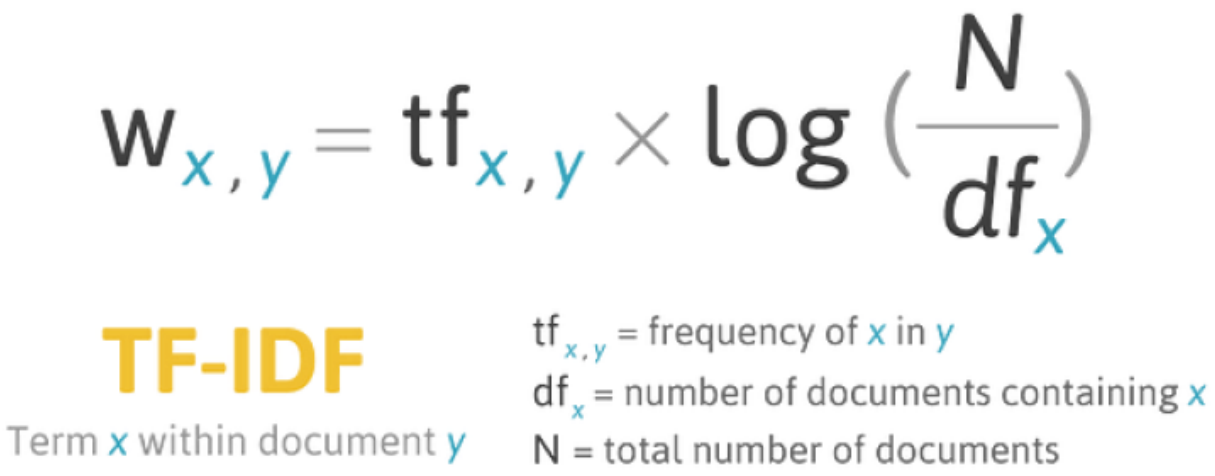


Рисунок 2.1 – Формула TF-IDF

Перевага TF-IDF у тому, що він не просто фіксує присутність слова, а оцінює його важливість у контексті всього корпусу. У задачі виявлення токсичності це дозволяє моделі краще відрізняти нейтральні тексти від таких, що містять специфічну лексику з негативним забарвленням. Наприклад, слова на кшталт «дурень», «огидний», «зрадник» можуть мати високу TF у токсичних коментарях, але низьку частоту в загальній вибірці, що робить їх вагомими предикторами класу. Водночас слова на кшталт «ти», «я», «це» будуть мати низький IDF, бо є характерними для будь-якого тексту, а отже – поганими ознаками.

TF-IDF реалізується як розріджена матриця, де кожен рядок – окремий документ (коментар), а кожен стовпчик – унікальне слово з усієї словникової бази. Ця матриця зазвичай має високу розмірність, проте

зберігається ефективно завдяки тому, що більшість коментарів містить лише невелику кількість слів зі всього корпусу. Саме це дозволяє ефективно поєднувати TF-IDF з простими класифікаторами, які не потребують складних нейронних структур.

На додачу до TF-IDF часто використовують n-грамні ознаки, які враховують послідовності слів. Наприклад, біграми (n=2) можна фіксувати вирази типу «ти дурень» або «ненавиджу тебе», які поодинці могли б не мати токсичного значення. Застосування трі- чи навіть чотири-грам дозволяє ще точніше виявляти шаблонні фрази, типові для агресивного спілкування. N-грамні ознаки особливо корисні при маскуванні токсичності: наприклад, слово «вб*ю» може обійти простий словник, але збережеться у відповідній n-грамі. Такі послідовності також можуть виявити саркастичні або пасивно-агресивні конструкції, що втрачаються при ізольованому аналізі слів.

Втім, при збільшенні n різко зростає кількість можливих комбінацій, що веде до розширення простору ознак і може викликати перенавчання, особливо якщо корпус невеликий. Тому при використанні n-грам зазвичай обмежуються n = 2 або 3, а словник додатково фільтрується за частотністю, аби уникнути шуму. Комбінація TF-IDF з n-грамами створює більш багатий опис тексту і є стандартною практикою у класичних підходах до NLP-класифікації [18].

Векторні подання на основі TF-IDF або n-грам формують основу для подальшої класифікації, яку виконують алгоритми машинного навчання. Класичні лінійні моделі добре працюють у таких задачах, оскільки можуть ефективно обробляти розріджені ознаки високої розмірності та виявляти ключові маркери класу. Найчастіше для цього застосовують логістичну регресію, метод опорних векторів (SVM) і наївний байєсівський класифікатор. Усі ці алгоритми мають давню історію застосування в текстовій аналітиці та залишаються актуальними завдяки стабільності, передбачуваності та невибагливості до ресурсів.

Логістична регресія – це лінійний класифікатор, який оцінює ймовірність належності прикладу до певного класу на основі зваженої суми ознак. Вона особливо добре підходить для розріджених векторів, оскільки ефективно виявляє релевантні ознаки навіть за великої кількості шуму. У задачах виявлення токсичності логістична регресія демонструє хорошу точність на невеликих або помірних корпусах і легко масштабується на багатокласові або мультиміткові випадки. До її сильних сторін належать простота інтерпретації (можна аналізувати ваги ознак) і невеликий ризик перенавчання, особливо при використанні регуляризації.

Метод опорних векторів (SVM) є потужнішим класифікатором, який шукає гіперплощину, що максимально розділяє класи (рисунок 2.2) [19]. Для текстових даних, які мають дуже високу розмірність, SVM із лінійним ядром зазвичай працює ефективніше, ніж нелінійні варіанти. Його ключова перевага – здатність зосередитись на «межових» прикладах, що часто відіграють вирішальну роль у токсичних діалогах: наприклад, коли фраза є сумнівною або двозначною. SVM зазвичай демонструє високу точність у задачах, де класи не повністю збалансовані, хоча його інтерпретація менш прозора, ніж у логістичній регресії.

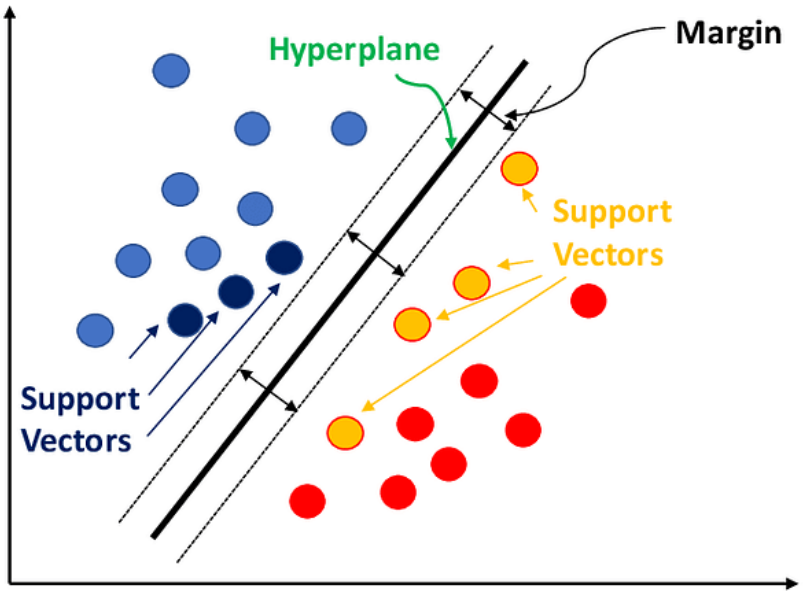


Рисунок 2.2 – Геометрична інтерпретація роботи методу опорних векторів

Наївний байєсівський класифікатор базується на обчисленні апостеріорних ймовірностей класів з використанням теореми Байєса, при цьому передбачається умовна незалежність усіх ознак [20]. Незважаючи на спрощене припущення, цей метод часто показує хороші результати в текстових задачах, де взаємна незалежність слів є наближеною, але прийнятною. Він особливо добре працює з невеликими наборами даних і має найнижчу обчислювальну складність серед трьох згаданих підходів. Байєсівські моделі рідко перевершують SVM або логістичну регресію в точності, але залишаються надзвичайно корисними як базові моделі або частина ансамблевих підходів.

Кожен з описаних алгоритмів має свої переваги залежно від характеристик даних: розмір корпусу, баланс класів, наявність шуму. У контексті виявлення токсичності класичні моделі все ще активно використовуються як на практиці, так і в експериментальних дослідженнях – зокрема, як початкові базові лінії для оцінки складніших нейронних підходів.

Основна перевага описаних методів – простота реалізації та інтерпретованість. Можна без значних ресурсів побудувати модель, яка здатна досить точно класифікувати короткі повідомлення, особливо якщо токсичність виражена прямим чином. Крім того, класичні моделі демонструють стабільність навіть при обмеженій кількості навчальних прикладів і не вимагають великих обчислювальних потужностей. Саме тому вони часто використовуються як базова точка порівняння в більш складних експериментах.

Однак такі моделі обмежені у своїх можливостях. Вони не враховують порядок слів, не розуміють контекст, не вловлюють іронію чи приховані змісти. У випадку неоднозначних або стилістично складних висловлювань, ймовірність помилкової класифікації суттєво зростає. Крім того, класичні ознаки мають обмежену здатність до узагальнення: неочікуване

формулювання, нове слово або замаскована токсичність можуть легко бути проігноровані.

Попри це, класичні ML-моделі залишаються цінним інструментом, особливо для побудови прототипів або для використання в задачах, де ресурси на глибоке навчання обмежені. У комплексних системах вони можуть застосовуватись як частина ансамблю або як попередній шар фільтрації перед запуском глибших моделей.

2.1.3 Глибокі нейронні мережі

Глибокі нейронні мережі стали наступним етапом у розвитку моделей для виявлення токсичності, коли обмежень класичних підходів виявилося недостатньо. На відміну від TF-IDF або n-грам, які розглядають текст як набір окремих елементів, глибинні моделі дозволяють працювати з послідовностями та вловлювати семантичні й синтаксичні зв’язки між словами. Завдяки цьому такі архітектури здатні розпізнавати складні лінгвістичні структури, приховану агресію, іронію, багатозначні висловлювання – тобто саме ті речі, які часто не вловлюються через просту частотну модель.

Однією з перших і найбільш впливових архітектур для роботи з послідовними даними стала LSTM (Long Short-Term Memory) – модифікована версія класичної рекурентної нейронної мережі (RNN), спеціально розроблена для подолання проблеми зникнення градієнту під час навчання на довгих послідовностях. В основі LSTM лежить ідея керованого збереження та оновлення інформації у вигляді внутрішнього стану пам’яті (cell state), який передається через часові кроки з мінімальними змінами, якщо це необхідно.

Архітектура LSTM (рисунок 2.3) складається з трьох головних елементів:

– forget gate f_t: визначає, яка частина попереднього стану C_{t-1} повинна бути збережена;

– input gate i_t: контролює, яка нова інформація $\tilde{C}_t$ буде додана до пам'яті;

– output gate o_t: регулює, яка частина оновленої пам'яті буде використана для формування вихідного стану h_t.

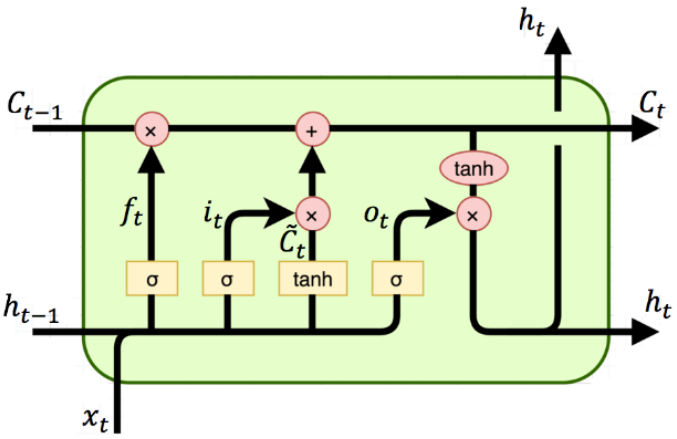


Рисунок 2.3 – Архітектура LSTM [21]

Ця гнучка структура дає змогу LSTM-мережам захоплювати як короткотривалі, так і довготривалі залежності, що критично важливо в завданнях аналізу коментарів, де токсичність може накопичуватися поступово або проявлятися лише в контексті попередніх фраз. Проте LSTM мають і свої недоліки: значна кількість параметрів призводить до високих обчислювальних витрат, а послідовний характер обробки ускладнює паралельне навчання, що знижує ефективність при масштабуванні на великі корпуси даних.

Як швидша та легша альтернатива була запропонована архітектура GRU (Gated Recurrent Unit). Вона спрощує структуру LSTM шляхом об'єднання forget і input воріт в єдине update gate z_t, а також введення reset gate r_t, що дозволяє контролювати, скільки попередньої інформації потрібно враховувати при генерації нового прихованого стану $\tilde{h}_t$ (рисунок 2.4).

GRU-мережі:

– ефективніші обчислювально, оскільки містять менше параметрів;

– швидше навчаються, що робить їх привабливими для практичного використання;

– у більшості випадків демонструють аналогічну або навіть кращу якість, ніж LSTM, особливо при обмежених обсягах навчальних даних.

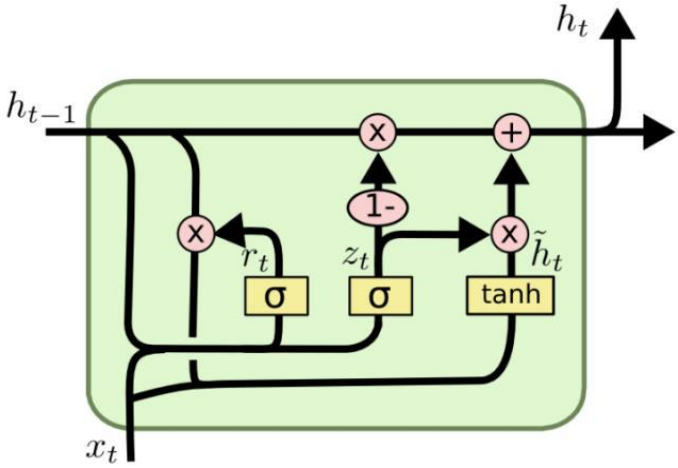


Рисунок 2.4 – Архітектура GRU [21]

Таким чином, вибір між LSTM і GRU залежить від компромісу між точністю, обчислювальними ресурсами та обсягом даних. У задачах виявлення токсичності, де важливий контекст та послідовність, обидві архітектури показують хороші результати. Проте GRU може бути доцільнішим варіантом у мобільних або онлайн-системах, де важлива швидкість відповіді та енергоефективність.

Іншим ефективним підходом до класифікації текстів є використання згорткових нейронних мереж (CNN). Попри те, що CNN спочатку були створені для аналізу зображень, їх успішно адаптовано для роботи з послідовностями тексту, зокрема завдяки здатності виявляти локальні ознаки – фрази чи словосполучення, які часто сигналізують про наявність токсичних патернів.

На рисунку 2.5 наведено типову архітектуру CNN для обробки текстів:

– Embedding layer – перетворює кожне слово у вектор фіксованої розмірності, зазвичай за допомогою попередньо навчених векторів (наприклад, Word2Vec, GloVe, FastText);

– Convolutional layer – застосовує фільтри різної висоти (від 2 до 5 слів), які ковзають по матриці ембедінгів і виявляють локальні шаблони. Це можуть бути, наприклад, образливі фрази або комбінації слів, характерні для певної форми агресії;

– Max-pooling layer – виділяє найбільш активні ознаки, дозволяючи зберегти найінформативніші локальні патерни незалежно від їх позиції в реченні;

– Concatenation layer – об’єднує ознаки з різних фільтрів, формуючи вектор фіксованої довжини;

– Softmax (або інший класифікатор) – здійснює остаточне передбачення класу (наприклад, «токсичний» / «нетоксичний»).

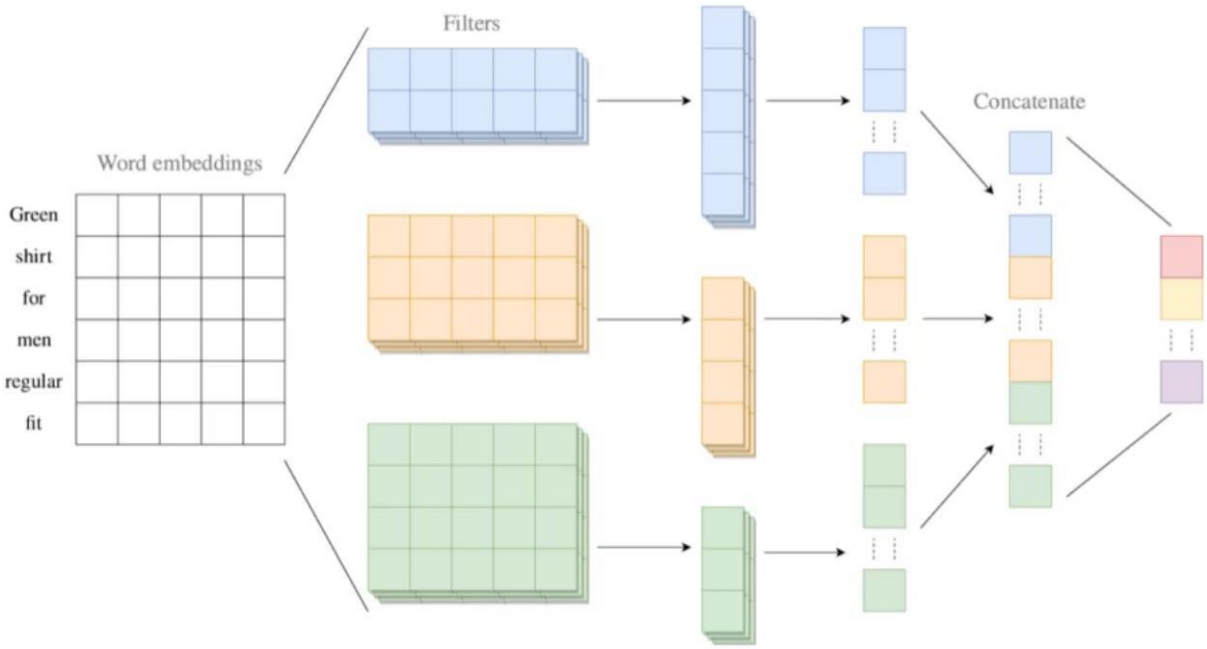


Рисунок 2.5 – Архітектура CNN [22]

Згорткові нейронні мережі мають низку практичних переваг, які роблять їх особливо корисними в задачах класифікації коротких текстів, зокрема у виявленні токсичних коментарів. Одна з ключових переваг полягає в їхній здатності швидко та ефективно обробляти вхідні послідовності завдяки локальності операцій. На відміну від рекурентних мереж, що обробляють дані поелементно в часовому порядку, CNN можуть одночасно застосовувати фільтри до всіх частин тексту, що суттєво прискорює як етап тренування, так і інференсу.

Також важливо відзначити, що CNN добре виявляють характерні локальні патерни у тексті – наприклад, типові образливі конструкції або специфічні словосполучення, які часто є маркерами токсичності. Ці шаблони можуть з’являтися в будь-якому місці репліки, і CNN залишаються до них чутливими незалежно від їхньої позиції, завдяки механізму згортки й наступного максимального пулінгу. При цьому модель не надто залежить від глобального порядку слів, що може бути перевагою в умовах варіативної побудови фраз в інтернет-комунікації.

Ще однією сильною стороною CNN є їхня стабільність у роботі з короткими текстами. Саме у випадках із лаконічними, емоційно насиченими коментарями – такими як твіти чи відповіді у чаті – згорткові мережі демонструють конкурентну точність, часто не поступаючись складнішим рекурентним архітектурам. Завдяки цьому вони особливо привабливі для застосування у мобільних додатках або інтерактивних веб-сервісах, де важлива швидкість відповіді та компактність моделі.

Попри свою потужність, глибокі моделі вимагають значних обсягів розмічених даних для навчання. Вони також менш прозорі в інтерпретації: важко визначити, чому саме певний коментар класифікується як токсичний. Це ускладнює їхнє використання в системах, де важлива пояснюваність або де є потреба в модераторському втручанні. У більшості практичних застосувань ці архітектури або замінені трансформерами, або використовуються як їхні складові у гібридних підходах. Проте як

перехідний етап до сучасних моделей, LSTM, GRU і CNN суттєво розширили можливості систем для обробки природної мови, і в окремих випадках залишаються доречними завдяки своїй відносній простоті.

2.1.4 Моделі на основі трансформерів

Моделі на основі трансформерів стали ключовим проривом у задачах обробки природної мови, зокрема – у виявленні токсичності. Їхня поява ознаменувала відмову від послідовного оброблення тексту (як у LSTM/GRU) на користь повністю паралельної архітектури, що аналізує контекст усієї послідовності одночасно. Завдяки механізму self-attention трансформери здатні враховувати зв’язки між будь-якими словами в тексті, незалежно від їхнього розташування, що особливо важливо для виявлення токсичності, яка може бути прихованою, відкладеною в часі або завуальованою.

Серед найуспішніших трансформерних архітектур у сфері обробки природної мови вирізняється BERT (Bidirectional Encoder Representations from Transformers). Її ключовою перевагою є здатність обробляти текст у двох напрямках: кожне слово аналізується з урахуванням як попереднього, так і наступного контексту. Це дозволяє моделі краще розуміти семантичні зв’язки у складних висловлюваннях, що суттєво покращує якість виконання широкого спектра завдань – від класифікації до відповіді на запитання.

Однак BERT є доволі ресурсоємною моделлю. Її архітектура передбачає наявність великої кількості параметрів і глибоку послідовну обробку, що підвищує обчислювальну складність як на етапі навчання, так і при використанні. Щоб подолати ці обмеження, було запропоновано кілька оптимізованих варіантів (рисунок 2.6).

Одним із таких є ALBERT (A Lite BERT) – модель, яка повторно використовує ті самі параметри на кожному рівні енкодера. Такий підхід дозволяє значно скоротити кількість параметрів без втрати архітектурної

потужності. Крім того, ALBERT застосовує факторизацію ембедінгів, що додатково знижує навантаження на пам’ять. У результаті отримано модель, здатну до глибокого семантичного аналізу при менших обсягах пам’яті.

Наступним кроком у цьому напрямі стала модель ELBERT, яка поєднує переваги ALBERT з механізмом раннього виходу (early exit). Вона передбачає, що модель може завершити обробку тексту на одному з проміжних шарів, якщо досягнуто достатнього рівня впевненості у результаті. Це дозволяє динамічно скорочувати глибину обчислень залежно від складності вхідного тексту, що особливо корисно в реальних застосунках із вимогами до часу відповіді або обмеженими обчислювальними ресурсами.

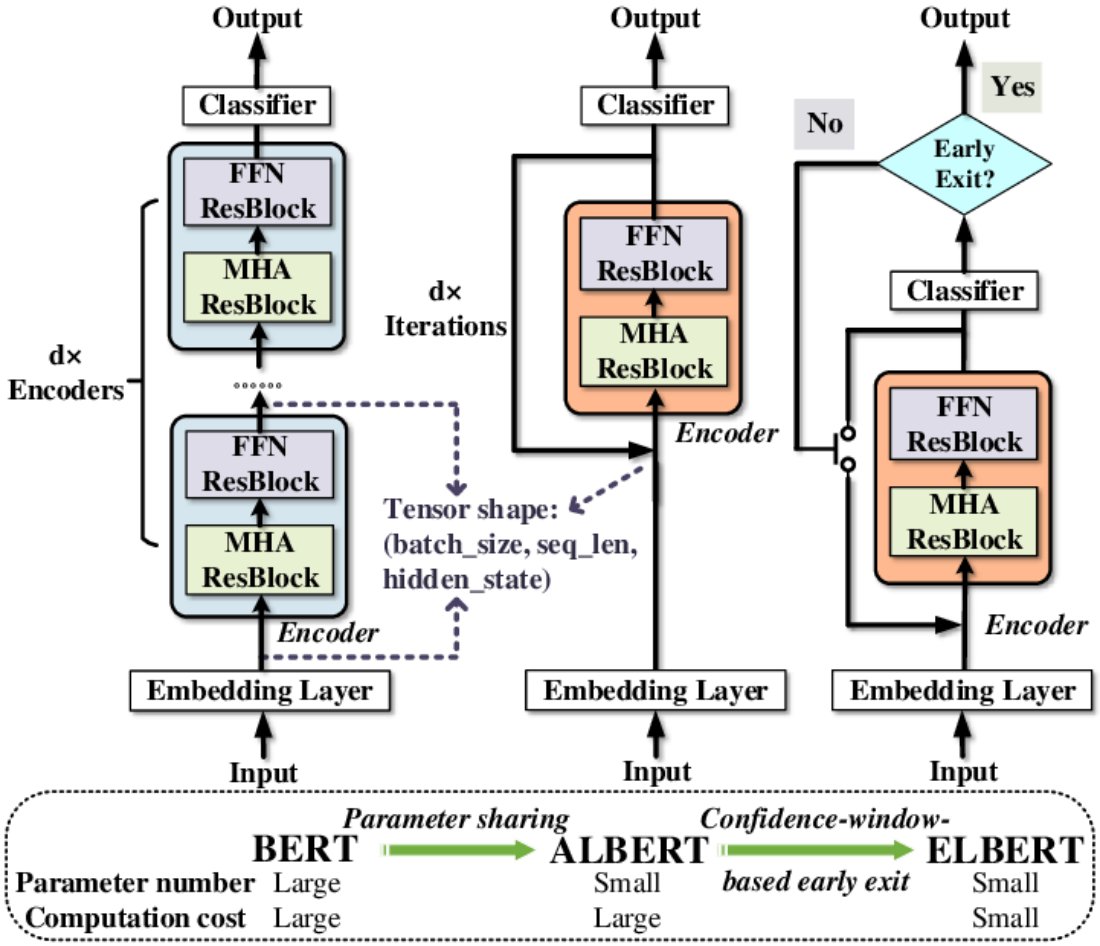


Рисунок 2.6 – Порівняння архітектур BERT, ALBERT і ELBERT

У задачі класифікації токсичності BERT використовується або у вигляді предтренованої моделі з доданим класифікаційним шаром, або як основа для донавчання (fine-tuning) на спеціалізованих датасетах, наприклад, Jigsaw Toxic Comment Classification.

Варто також згадати про RoBERTa – покращену версію BERT з модифікованими параметрами навчання, яка демонструє ще кращі результати на низці NLP-задач, у тому числі токсичності. Завдяки масштабнішому тренуванню, відсутності NSP-завдання та динамічному маскуванню, RoBERTa краще узагальнює складні семантичні структури й виявляє непрямі форми агресії. Модель DistilBERT, у свою чергу, забезпечує компроміс між точністю та швидкістю: вона менш ресурсоємна, але придатна для розгортання в реальному часі або на менш потужному обладнанні.

Крім загальних моделей, існують спеціалізовані рішення, адаптовані для задач виявлення токсичності. Серед них – Detoxify, що базується на RoBERTa та fine-tuned на Jigsaw-коментарях із різними класами агресії; а також моделі від Jigsaw/Perspective API, які надають API-доступ до предтренованих класифікаторів токсичності в кількох мовах. Для української мови варто згадати xlm-roberta-large-uk-toxicity, яка навчається на багатомовному корпусі з додатковими українськими токсичними прикладами й дає змогу будувати локалізовані моделі без необхідності ручного перекладу корпусів.

Завдяки високій точності, гнучкості та здатності вловлювати глибинні смислові зв’язки, трансформери стали сучасним стандартом у задачах виявлення токсичності. Вони ефективно обробляють як прості, так і контекстно складні висловлювання, демонструють хорошу узагальнюваність і дають змогу працювати у zero-shot або few-shot режимах, що особливо актуально для багатомовного середовища або обмеженого набору міток.

2.2 Підходи до генерації текстових відповідей

Генерація відповідей на текстові повідомлення – окрема категорія задач в обробці природної мови, яка відрізняється від класифікації як за метою, так і за рівнем складності. Якщо класифікатор має лише визначити, чи містить коментар токсичність, то генератор мусить побудувати змістовну, зв’язну й стилістично відповідну репліку. До цього завдання застосовуються різні підходи – від класичних моделей типу seq2seq до сучасних трансформерних систем з гнучким управлінням тоном і змістом. Вибір методу залежить не тільки від точності, а й від того, наскільки модель здатна контролювати стиль, враховувати контекст і не генерувати небажаних чи двозначних відповідей.

2.2.1 Seq2Seq на основі RNN

Одним із перших ефективних підходів до генерації тексту став seq2seq (sequence-to-sequence), побудований на основі рекурентних нейронних мереж (RNN). Архітектура такого типу складається з двох частин: енкодера, який зчитує вхідну послідовність і кодує її у внутрішнє представлення (вектор фіксованої довжини), та декодера, який по цьому вектору поетапно генерує нову послідовність – тобто відповідь. Цей підхід дозволяє моделі адаптивно будувати текст, відштовхуючись від конкретного запиту, а не за шаблоном, і став основою для перших генеративних чат-ботів та перекладачів.

У базовій формі seq2seq на RNN мав значні обмеження: модель не могла зберігати довгі контексти, а кожне слово вихідного тексту залежало лише від попереднього, що часто призводило до втрати змісту або одноманітних відповідей. З появою механізму attention ситуація змінилася: декодер отримав можливість «зазирнути» в різні частини вхідної послідовності під час генерації кожного слова. Це дозволило значно

покращити якість побудови відповіді та зробило можливим використання seq2seq у діалогових системах. Модифікації на основі LSTM або GRU з attention і зараз залишаються робочим варіантом для простих генераторів (рисунок 2.7).

У задачах етичного реагування seq2seq дає змогу створювати короткі, стислі відповіді, що стилістично нагадують людські, однак при цьому модель лишається чутливою до помилок у структурі діалогу. Вона може не вловити підтекст, не розпізнати стиль вхідного повідомлення або побудувати загальну відповідь, яка не враховує тональність запиту. Крім того, як і всі RNN-архітектури, seq2seq не піддається ефективному паралельному обчисленню, що ускладнює масштабування і використання в реальному часі.

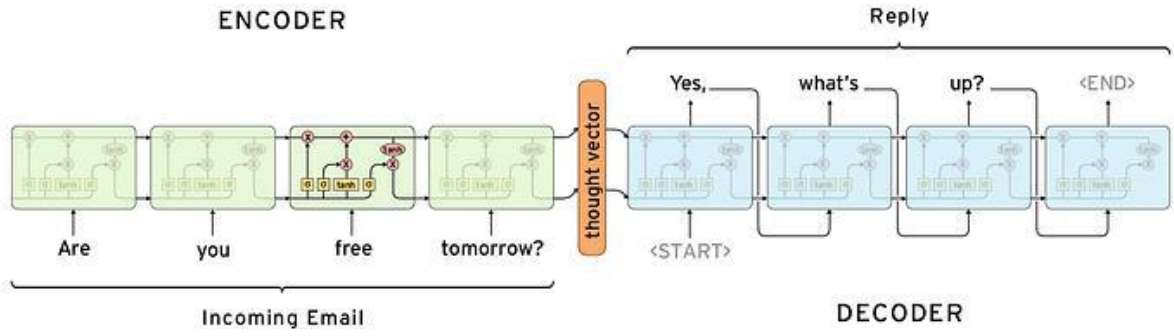


Рисунок 2.7 – Архітектура Seq2Seq з LSTM-енкодером і декодером для генерації відповіді [24]

Попри свої обмеження, seq2seq з attention усе ще використовується як легка альтернатива трансформерам у задачах генерації коротких реплік. Його головна перевага – простота налаштування, мінімальні вимоги до ресурсів і хороша відтворюваність результатів при навчанні на вузькоспеціалізованих корпусах. У деяких випадках, особливо за умов браку даних або при потребі в інтерпретованості, seq2seq може бути кращим за більш складні моделі.

2.2.2 Трансформери для генерації

Впровадження трансформерних архітектур повністю змінило підхід до генерації тексту. На відміну від послідовних моделей на основі RNN, трансформери оперують усією вхідною послідовністю одночасно завдяки механізму self-attention. Це дозволяє моделі враховувати глобальний контекст і підтримувати зв’язність на довгих ділянках тексту без втрат пам’яті чи накопичення помилок. Саме завдяки цим властивостям трансформери стали основою для сучасних генеративних систем, які не лише створюють граматично правильні тексти, а й здатні контролювати стиль, зміст і навіть емоційне забарвлення відповіді.

Однією з найвідоміших генеративних архітектур є GPT (Generative Pre-trained Transformer). Вона побудована за принципом автогресивної генерації, коли кожне наступне слово передбачається на основі вже згенерованої послідовності. Такий підхід дозволяє моделі поступово розвивати думку, створюючи узгоджений і логічно пов’язаний текст, а також чутливо реагувати на зміни контексту навіть у довгих діалогах.

На рисунку 2.8 зображено загальну архітектуру GPT-моделі: від вхідного шарування (ембедінгів і позиційного кодування) до багаторівневого стеку трансформерних блоків з механізмами самоуваги, нормалізації та нелінійних перетворень. Кожен блок працює над тим, щоб посилити контекстне розуміння вже згенерованої частини послідовності, і на цій основі формувати ймовірне наступне слово. На фінальному етапі застосовується softmax-шар, який обирає найімовірніший варіант наступного токена.

GPT виявилася надзвичайно ефективною у сфері діалогових систем, зокрема – в моделюванні емпатійних, стилізованих або ввічливих відповідей. Її сильна сторона полягає в гнучкості: якість і стиль відповіді значною мірою залежать від формулювання запиту (prompt), що дозволяє керувати тоном, формальністю або навіть емоційністю тексту без потреби

змінювати саму модель. Саме тому GPT стала основою багатьох сучасних систем, які не лише генерують граматично коректний текст, а й адаптуються до інструкцій користувача.

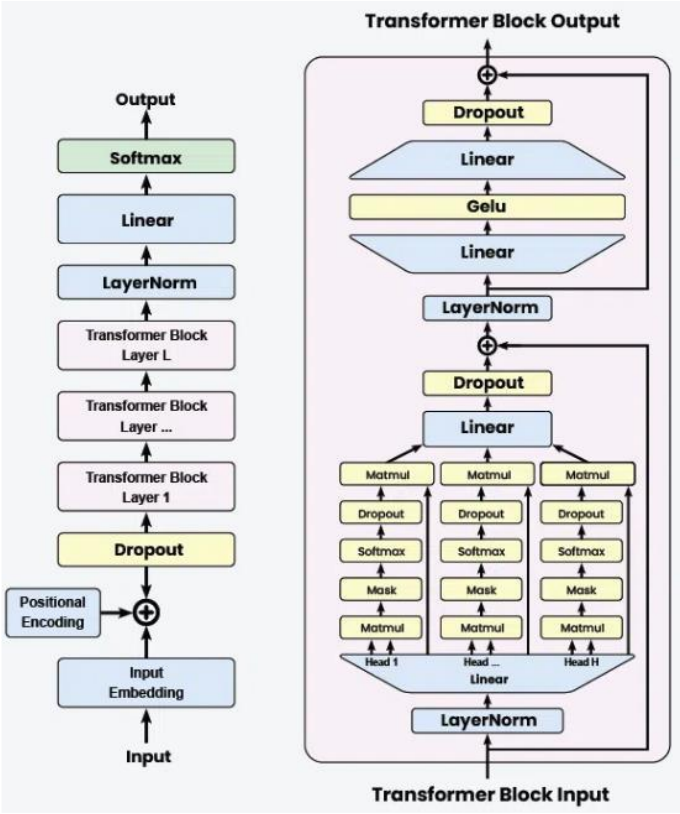


Рисунок 2.8 – Архітектура генеративного трансформера GPT [25]

Інша популярна архітектура – T5 (Text-To-Text Transfer Transformer) – реалізує класичну схему encoder-decoder, у якій вхідна послідовність повністю кодується, а потім поетапно декодується у відповідь. Така структура особливо ефективна в задачах переформулювання, узагальнення, перекладу або генерації на основі інструкцій. На відміну від GPT, яка працює в автогресивному режимі з урахуванням лише попереднього контексту, модель T5 має доступ до всієї вхідної інформації одразу, що дозволяє отримувати більш контрольовані й послідовні результати.

На рисунку 2.9 представлено архітектуру трансформера з енкодером та декодером, яку реалізує модель T5. Кожен блок енкодера містить шари

самоуваги, нормалізації та feed-forward перетворення. Декодер, у свою чергу, додатково включає механізм «encoder-decoder attention», який забезпечує зв’язок між вхідною та вихідною послідовностями, дозволяючи моделі фокусуватися на релевантних частинах вхідного тексту під час генерації відповіді.

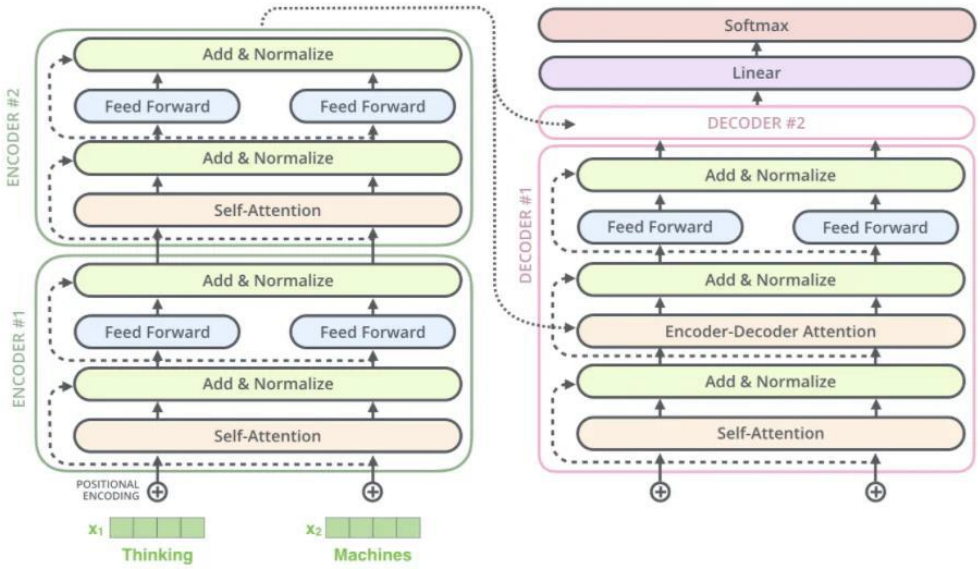


Рисунок 2.9 – Архітектура моделі T5 з encoder–decoder структурою [26]

Ще однією важливою особливістю T5 є уніфікований підхід до задач формулювання тексту: модель навчається виконувати широкий спектр завдань (класифікація, переклад, відповіді на запитання тощо) у єдиному форматі, де все розглядається як задача трансформації одного тексту в інший. Такий підхід підвищує гнучкість і дозволяє легко адаптувати модель до нових задач без потреби змінювати архітектуру або механізм тренування.

Головною перевагою трансформерів у генерації є здатність масштабуватися – як у розмірі моделей, так і в кількості даних. Вони працюють ефективно у few-shot та zero-shot режимах, тобто можуть генерувати якісні відповіді навіть без додаткового донавчання, лише на основі інструкції. Це особливо цінно в задачах етичного реагування, де складно зібрати великий розмічений корпус стилізованих реплік. Крім того,

трансформери відкривають простір для використання керованої генерації: моделі навчаються не лише на самих відповідях, а й на стилях, інтонаціях, соціальних сценаріях.

Попри свої переваги, трансформери залишаються вимогливими до ресурсів і складними в адаптації. Щоб використовувати їх ефективно, необхідно продумати промпти, стилістичні обмеження й системи фільтрації результату. Проте саме ці моделі дозволяють реалізувати функціональну, контекстно чутливу систему реагування на токсичні коментарі без обмеження на заздалегідь задані шаблони.

2.2.3 Підходи до стилістичного контролю

У задачах автоматичної генерації відповідей, особливо коли йдеться про етичне реагування, недостатньо створити просто граматично правильний текст. Важливо, щоб відповідь відповідала обраному стилю – була нейтральною, ввічливою або м’яко іронічною. Для цього застосовуються різні методи стилістичного контролю, які дають змогу керувати інтонацією, тональністю та форматом генерованого повідомлення. Ці методи можуть реалізовуватись як на рівні архітектури моделі, так і на рівні запиту до неї [27].

Найпростішим і найпоширенішим є prompt engineering – формування вхідного запиту у спосіб, що спрямовує модель на бажаний стиль відповіді. Наприклад, додавання інструкцій на кшталт «respond politely» або «use a soft tone» змінює інтонацію навіть без зміни самої моделі. Цей підхід широко використовується з моделями GPT і T5, оскільки вони добре реагують на вербальні інструкції. Його перевага – повна відсутність потреби в донавчанні, але водночас і недолік – нестабільність: одна і та сама інструкція може дати різний результат залежно від формулювання та контексту.

Другий підхід – fine-tuning на стилізованих корпусах. У цьому випадку модель навчається не просто генерувати текст, а генерувати його в певному стилі, притаманному навчанню. Наприклад, якщо модель донавчена на відповідях з форуму техпідтримки або з дипломатичних листувань, вона автоматично відтворює притаманну цим джерелам тональність. Цей метод дає стабільні результати, але вимагає великої кількості якісно розмічених даних для кожного стилю окремо, а також значних обчислювальних ресурсів.

Інший підхід – використання control tokens або маркерів стилю, які подаються разом з вхідним текстом і сигналізують моделі бажану стилістичну властивість. Наприклад, на початку запиту можна вставити спеціальний токен [polite] або [neutral], і модель буде вчитись асоціювати його з відповідним шаблоном мовлення. Таке маркування зазвичай поєднується з fine-tuning, але потребує чіткої інструкції в даних. Існують також моделі, що інтегрують стилістичні атрибути у внутрішнє представлення, дозволяючи змінювати стиль відповіді без повного перенавчання.

Нарешті, в окремих випадках використовується postprocessing – модифікація вже згенерованого тексту через перефразування, заміну слів або вставку ввічливих конструкцій. Це дозволяє «пом’якшити» грубий або неетичний варіант, однак має обмежену точність і не гарантує стилістичної цілісності. Постобробка може бути корисною як додатковий фільтр, але не є самостійним рішенням для створення реплік у контрольованому стилі.

Комбінування вищеназваних підходів – стандартна практика у складних генеративних системах. Наприклад, система може одночасно використовувати prompt з інструкцією, подавати стилістичний токен і застосовувати фільтр на виході. Це дає змогу досягти вищої стабільності та точності у відтворенні бажаного ефекту – зокрема, в системах, що мають працювати у відкритому середовищі з непередбачуваними запитами.

3 ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ СИСТЕМИ

3.1 Середовище розробки

Розробку програмної системи було здійснено мовою Python 3.10 у середовищі Visual Studio Code. Для ізоляції залежностей використовувалося віртуальне середовище venv, що забезпечило стабільну роботу з великими бібліотеками для обробки природної мови, моделювання та генерації тексту.

Основу технологічного стеку становили бібліотеки Torch та Transformers від HuggingFace. Саме вони забезпечили доступ до попередньо натренованих моделей roberta-base та xlm-roberta-large, які були використані для класифікації токсичних коментарів англійською та українською мовами відповідно. Крім того, інтеграція з сервісом OpenAI дозволила залучити модель GPT-4o для генерації етичних відповідей на виявлені токсичні висловлювання.

Інструмент langdetect використовувався для автоматичного визначення мови введеного тексту, що дозволило коректно маршрутизувати повідомлення до відповідного класифікатора. Для побудови користувацького інтерфейсу було обрано Streamlit – мінімалістичний веб-фреймворк, який дав змогу швидко створити веб-додаток без складної front-end розробки.

Проєкт мав модульну структуру (рисунок 3.1), що дозволило логічно розмежувати частини відповідальні за навчання, інференс, генерацію відповідей, обробку даних та візуалізацію результатів. У каталозі classifier_en розміщено повний стек навчання англомовного класифікатора, включно з графіками, підготовкою даних та аналізом помилок. Всі функції генерації етичних відповідей зібрано у generator_en, зокрема реалізацію prompt-інженерії та стилістичного контролю. Головний пайплайн для класифікації та генерації розміщено у файлі toxicity.py, а окремо реалізовано інтерактивний інтерфейс у app.py.

```
toxicity/
├── toxicity.py
├── app.py
├── classifier_en/
│   └── model/
├── classifier_ukr/
├── generator_en/
│   ├── generator_gpt.py
│   └── prompts.py
├── toxic_comments.csv
├── requirements.txt
├── .streamlit/config.toml
```

Рисунок 3.1 – Структура проєкту

3.2 Опис вхідних даних

Для навчання англомовної моделі класифікації токсичних коментарів було використано відкритий датасет Jigsaw Toxic Comment Classification Challenge, опублікований на платформі Kaggle [28]. Цей набір містить понад 160 000 англомовних коментарів з обговорень Wikipedia, які було вручну розмічено на наявність різних форм токсичності.

Оригінальний набір має мульти-лейблову структуру з такими категоріями, як toxic, severe_toxic, obscene, threat, insult, identity_hate. Оскільки метою цієї роботи є виявлення токсичних повідомлень загалом, усі мітки було зведено до однієї бінарної:

– 1 (toxic) – якщо принаймні один токсичний тег дорівнює 1;

– 0 (non-toxic) – якщо всі теги дорівнюють 0.

Після цього з усього набору було випадково вибрано по 5000 прикладів кожного класу, щоб сформувати збалансовану вибірку з 10 000 рядків. Очищення включало залишення лише тексту коментаря та створеної бінарної мітки. У фінальному вигляді датасет мав дві колонки – text та label (рисунок 3.2).

🔍 Sample comments:

text	label
There seems to exist a branch of this family in Finland nowadays....	non-toxic
Breakaway Repubic =/=Breakaway region What republic? There's only one: Republic of Moldova...	non-toxic
History of Eye Color article Seriously...not a good article. No offense, but could you sti...	non-toxic
":How do you think it looks that you are getting me blocked and then coming here to twist ...	toxic
MY FARTS DONT SMELL ou...	toxic
The only thing you are persistant at you ugly bastard is fucking the TROLL'''...	toxic

Рисунок 3.2 – Приклади коментарів із мітками toxic / non-toxic

Для візуального підтвердження балансування класів було побудовано стовпчикову діаграму, яка відображає рівну кількість прикладів кожного типу (рисунок 3.3).

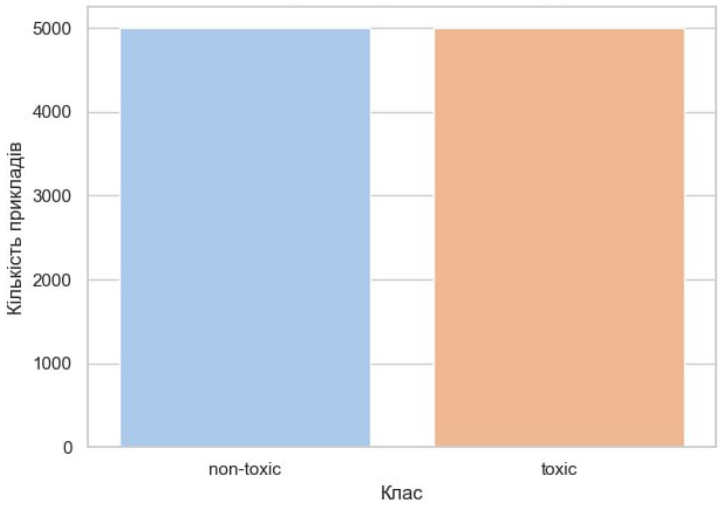


Рисунок 3.3 – Розподіл токсичних та нетоксичних коментарів

Перед тренуванням модель вимагала попередньої токенізації. Для цього було використано токенізатор RobertaTokenizer з моделі roberta-base. Вхідні тексти перетворювались у формат, прийнятний для трансформерної архітектури:

– input_ids – числові індекси токенів;

– attention_mask – маска уваги (1 для реальних токенів, 0 для паддінгу);

– label – цільова мітка.

Було встановлено максимальну довжину послідовності – 128 токенів, з обрізанням (truncation) і доповненням (padding), якщо потрібно. Токенізація здійснювалася пакетно за допомогою бібліотеки Hugging Face datasets, з подальшим збереженням у форматі PyTorch через .save_to_disk().

Після токенізації дані було поділено у співвідношенні 80:20 на навчальну та тестову вибірки, які зберігаються у classifier_en/data/train та classifier_en/data/test відповідно.

Для кращого розуміння, як саме текст перетворюється у вхідний формат моделі, на рисунку 3.4 наведено приклад токенізованого рядка, що включає input_ids, attention_mask і мітку.

text	input_ids	attention_mask
I hate you.	[0, 100, 4157, 47, 4, 2, 1, 1, 1, 1, 1, 1]	[1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0]
Thank you for your comment.	[0, 13987, 47, 13, 110, 1129, 4, 2, 1, 1, 1, 1]	[1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0]

Рисунок 3.4 – Приклад токенізованого тексту у форматі моделі RoBERTa

Для перевірки ефективності україномовної моделі класифікації токсичних повідомлень було створено власний контрольний набір даних, який складався зі 200 прикладів. Половина з них містила явну або приховану токсичність (лайка, образи, узагальнення, сарказм), тоді як інша половина – нейтральні або ввічливі формулювання. Таким чином, вибірка була збалансованою за класами (рисунок 3.5).

Початкові приклади було зібрано вручну, після чого виконано очистку за допомогою скрипту prepare_testset.py. У процесі підготовки видалялися зайві лапки, перенос рядків, надмірні пробіли та некоректні символи. Тексти також було приведено до типу str, а мітки – до типу Int64, що забезпечило коректне зчитування під час інференсу.

Остаточна версія датасету була збережена у вигляді файлу classifier_ukr/clean_test_samples.csv.

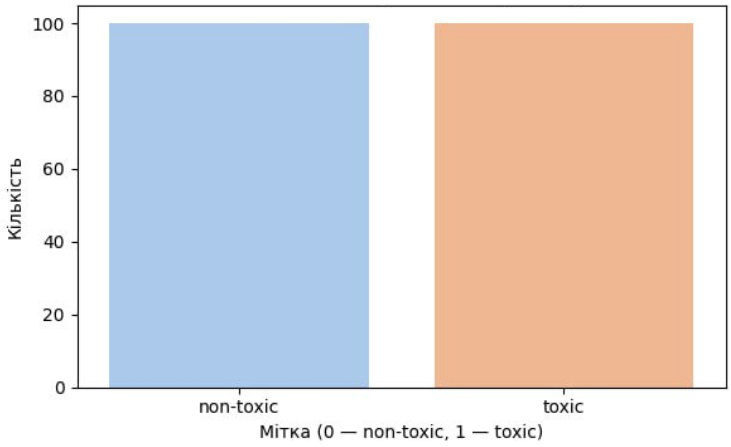
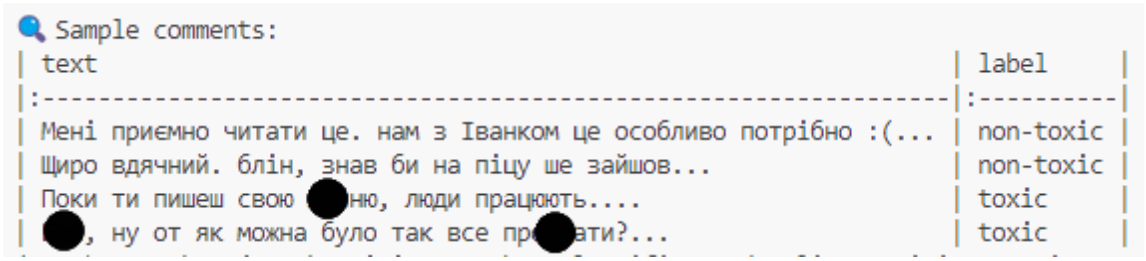


Рисунок 3.5 – Розподіл класів у контрольному наборі

Цей набір не призначався для навчання, а лише для тестування якості вже готової моделі. Формат файлу відповідав структурі англомовного датасету: кожен рядок містив текст повідомлення та відповідну мітку (рисунок 3.6).



🔍 Sample comments:

text	label
Мені приємно читати це. нам з Іванком це особливо потрібно :(...	non-toxic
Щиро вдячний. блін, знав би на піцу ше зайшов...	non-toxic
Поки ти пишеш свою [redacted]ню, люди працюють....	toxic
[redacted], ну от як можна було так все пр[redacted]ати?...	toxic

Рисунок 3.6 – Приклади коментарів із мітками toxic / non-toxic

3.3 Архітектура системи

Інтелектуальна система підтримки етичної комунікації побудована як послідовний обробник текстових повідомлень користувача. Її архітектура включає три взаємопов’язані компоненти: класифікацію токсичності, визначення мови повідомлення та генерацію етичної відповіді. Уся логіка реалізована у файлі toxicity.py.

На вхід система приймає довільний текстовий коментар. Після цього виконується визначення мови за допомогою бібліотеки langdetect. Це дозволяє автоматично вибрати відповідну модель класифікації – англомовну або україномовну:

– якщо мова англійська, система використовує модель roberta-base, яку попередньо донавчено на збалансованому наборі англомовних коментарів (5000 токсичних і 5000 нетоксичних);

– якщо мова українська, застосовується вже готова модель dardem/xlm-roberta-large-uk-toxicity, призначена для задачі бінарної класифікації токсичності.

Обидві моделі працюють через бібліотеку transformers, використовуючи відповідні токенізатори (RobertaTokenizer або AutoTokenizer). У процесі інференсу повідомлення токенізується з параметрами padding=True, truncation=True, max_length=128. Далі модель повертає логіти, з яких визначається клас: toxic або non-toxic.

Якщо повідомлення класифіковане як токсичне, система переходить до етапу генерації відповіді. Для цього викликається функція generate_response() з модуля generator_gpt.py, яка звертається через API OpenAI до моделі GPT-4o, що формує відповідь у стилі:

– neutral – інформативний, стриманий тон;

– polite – ввічлива мова з легким осудом;

– soft – гумор, емпатія, м’яка реакція.

Система підтримує два варіанти режиму роботи:

– генерація однієї відповіді у заданому стилі (наприклад, лише soft);

– генерація всіх трьох стилів одночасно з виведенням кожного окремо.

Кожен стиль реалізовано як шаблонований інструкційний промпт, адаптований до мови коментаря. Якщо коментар українською – промпт також формується українською. Це дозволяє уникнути перекладу на рівні системи і забезпечити природність відповідей.

Усі дії – введення коментаря, визначення мови, класифікація, виклик генератора – відбуваються в рамках єдиного сценарію, що запускається з консолі. Повідомлення Prediction: toxic або non-toxic супроводжується відповідями у заданих стилях (якщо активовано ENABLE_RESPONSE=True).

Архітектура системи передбачає мінімальні залежності, що дозволяє розгорнути її локально на будь-якій машині з підтримкою Python, PyTorch і OpenAI API.

3.4 Класифікатор токсичності

Виявлення токсичних повідомлень є ключовим етапом у роботі системи підтримки етичної комунікації. На цьому етапі вхідний текст аналізується класифікатором, який визначає, чи порушує повідомлення норми етичної взаємодії. У рамках проєкту реалізовано два окремі модулі для класифікації: один обробляє англомовні повідомлення, інший – україномовні.

Обидва класифікатори використовують трансформерні архітектури, натреновані на відповідних даних. Англомовна модель була донавчена на збалансованій підмножині датасету Jigsaw, тоді як для української використовувалася вже готова модель, натренована на корпусі україномовних коментарів. Усі рішення про токсичність приймаються у бінарній формі: toxic або non-toxic.

3.4.1 Англомовна модель

Для класифікації токсичних англомовних повідомлень була використана модель roberta-base, яка є попередньо натренованою трансформерною архітектурою. Модель було донавчено на власному бінаризованому датасеті, описаному у розділі 3.2. Усього в тренувальній

вибірці було 10 000 прикладів (5000 токсичних і 5000 нетоксичних), з яких 80% використовувалися для навчання, а 20% – для тестування.

Навчання відбувалося за допомогою бібліотеки transformers у режимі Trainer. Вхідні дані токенізувалися токенізатором RobertaTokenizer з максимальною довжиною 128 токенів. Було використано оптимізатор AdamW, епоха тренування – 2, learning rate – 2e-5, batch size – 8. Обрана метрика для відбору найкращої моделі – f1-score.

Конфігурація моделі передбачає понад 124 мільйони параметрів, з яких усі є тренованими. Класифікаційна голова складається з двох лінійних шарів і Dropout, що накладається після виходу з трансформера. Структуру моделі наведено на рисунку 3.7.

Layer (type (var_name))	Input Shape	Output Shape	Param #	Trainable
RobertaForSequenceClassification (RobertaForSequenceClassification)	--	[1, 2]	--	True
├─RobertaModel (roberta)	[1, 128]	[1, 128, 768]	--	True
│ └─RobertaEmbeddings (embeddings)	--	[1, 128, 768]	39,000,576	True
│ └─RobertaEncoder (encoder)	[1, 128, 768]	[1, 128, 768]	85,054,464	True
├─RobertaClassificationHead (classifier)	[1, 128, 768]	[1, 2]	--	True
│ └─Dropout (dropout)	[1, 768]	[1, 768]	--	--
│ └─Linear (dense)	[1, 768]	[1, 768]	590,592	True
│ └─Dropout (dropout)	[1, 768]	[1, 768]	--	--
│ └─Linear (out_proj)	[1, 768]	[1, 2]	1,538	True

Total params: 124,647,170
Trainable params: 124,647,170
Non-trainable params: 0
Total mult-adds (M): 124.65

Input size (MB): 0.00
Forward/backward pass size (MB): 106.96
Params size (MB): 498.59
Estimated Total Size (MB): 605.55

Рисунок 3.7 – Архітектура моделі RoBERTa для класифікації

Після завершення тренування було виконано оцінку на тестовій вибірці. Загальна точність класифікатора склала 0.9405, макроусереднене значення F1 – 0.9405, що вказує на високу збалансованість моделі по обох класах.

Детальний звіт з precision, recall, f1-score наведено на рисунку 3.8, а рисунок 3.9 демонструє матрицю помилок.

Classification report:

	precision	recall	f1-score	support
non-toxic	0.9565	0.9230	0.9394	1000
toxic	0.9256	0.9580	0.9415	1000
accuracy			0.9405	2000
macro avg	0.9410	0.9405	0.9405	2000
weighted avg	0.9410	0.9405	0.9405	2000

Рисунок 3.8 – Класифікаційний звіт

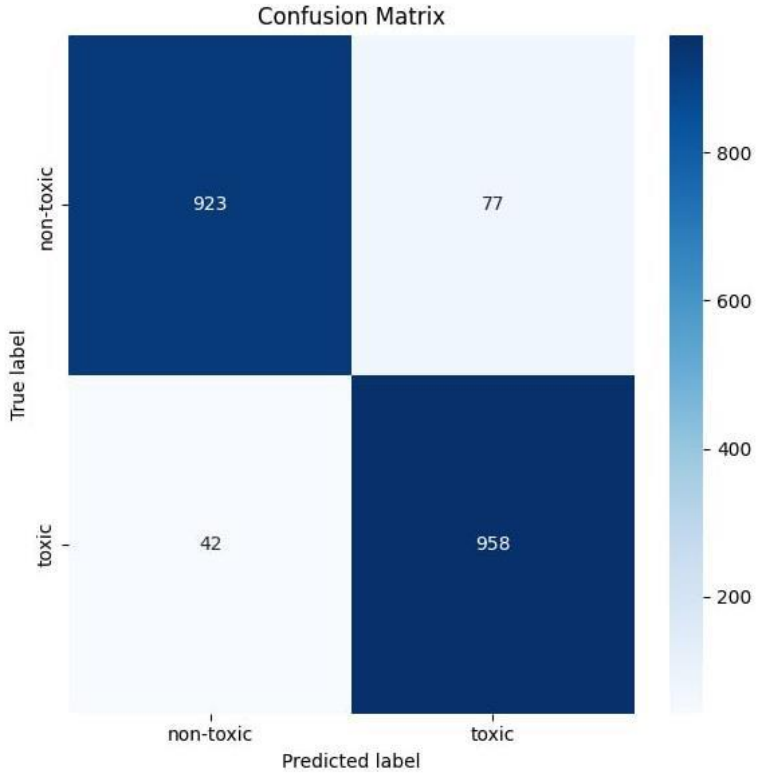


Рисунок 3.9 – Матриця помилок на тестовій вибірці

Отримані результати засвідчують, що донавчена roberta-base здатна ефективно розрізняти токсичний і конструктивний тон навіть на сильно скороченому датасеті.

Аналіз помилок (рисунок 3.10) дозволив виявити характерні слабкі місця моделі. Серед false positives (коли модель хибно вважала

повідомлення токсичним) переважали приклади, що мали емоційно забарвлений тон або соціально чутливу лексику. Наприклад:

"Just because you don’t see someone showing racial discrimination..."

"This is a democracy and I am expressing my views!"

У цих випадках не використовувалося образливої мови, однак присутній емоційний тиск або конфронтаційний настрій, який модель могла трактувати як сигнал токсичності.

Натомість false negatives (пропущені токсичні повідомлення) часто містили прямі погрози або звернення з домінантною позицією, але були подані без очевидної лайки. Наприклад:

"Do not tell me what to do or you will be blocked."

"Hey Dictator, how would you like to be on an online radio program for a debate?"

Ці приклади демонструють, що модель може втрачати токсичність, якщо вона подана у формально нейтральній граматичній конструкції або завуальована через іронію, сарказм чи риторичні прийоми.

False Positives (модель помилково класифікувала як токсичне):

short_text	true_label	predicted_label
Here goes... I'm not the greatest hook writer; someone else might have a sexier way to put it.	non-toxic	toxic
" (UTC) Just because you don't see someone showing racial discrimination doesn't mean someone ain'...	non-toxic	toxic
I'm fed up with all this and the people here. The community here is dying and now I will now indefin...	non-toxic	toxic
...AND ALSO DO NOT DELETE MY COMMENT, this is a democracy and i am expressin my views! you just dont...	non-toxic	toxic
" Awaaze read your pov "" 8000 cases of missing persons have been reported since the start of the w...	non-toxic	toxic

False Negatives (модель не розпізнала токсичність):

short_text	true_label	predicted_label
" No, Caspian, it is you and your mode of operation that is transparent. That is the big problem w...	toxic	non-toxic
" Gita’s Samkhya is NOT DIFFERENT ""Only the ignorant speak of devotional service [karma-yoga] a...	toxic	non-toxic
Warning	toxic	non-toxic
Do not tell me what to do or you will be blocked		
Courtesy of this phenomenon my son already has procured an arrest record.	toxic	non-toxic
Hey Dictator how would you like to be on an online radio program up for a debate? Its through the ph...	toxic	non-toxic

Рисунок 3.10 – Приклади помилкової класифікації: FP та FN

Загалом, модель roberta-base показала впевнене розрізнення класів у межах навчального домену, однак виявила чутливість до латентної емоційності та контекстної неоднозначності.

3.4.2 Україномовна модель

Для класифікації токсичності в україномовних повідомленнях використано модель dardem/xlm-roberta-large-uk-toxicity [29], що побудована на архітектурі XLM-RoBERTa-large – багатомовному трансформері, призначеному для високоякісного аналізу тексту низкою мов, включно з українською. Модель попередньо донавчена на спеціалізованому корпусі токсичних коментарів українською мовою, тож у межах цієї роботи її було застосовано без додаткового тренування.

Якість моделі оцінювалась на контрольному наборі з 200 прикладів. Оскільки цей набір був збалансованим (порівну токсичних і нетоксичних повідомлень), він дозволив отримати достовірну оцінку як precision, так і recall. Тестування проводилося за допомогою скрипту test_inference.py, який виконував токенізацію, обчислення логітів, визначення класів та підсумкову оцінку.

Підсумкова точність моделі склала 91%. Детальні результати класифікації наведено на рисунку 3.11. Модель краще розпізнавала нетоксичні повідомлення (recall = 0.98), проте мала вищу precision у класі toxic (0.98). Таким чином, система схильна з обережністю ставитися до класифікації тексту як токсичного.

Classification report:

	precision	recall	f1-score	support
non-toxic	0.8596	0.9800	0.9159	100
toxic	0.9767	0.8400	0.9032	100
accuracy			0.9100	200
macro avg	0.9182	0.9100	0.9096	200
weighted avg	0.9182	0.9100	0.9096	200

Рисунок 3.11 – Класифікаційний звіт для україномовної моделі

Результати підтверджуються і матрицею помилок (рисунок 3.12), яка демонструє, що з 200 прикладів лише 18 були класифіковані неправильно: 2 false positives (нейтральні повідомлення, помилково позначені як токсичні) та 16 false negatives (токсичні повідомлення, які не були розпізнані).

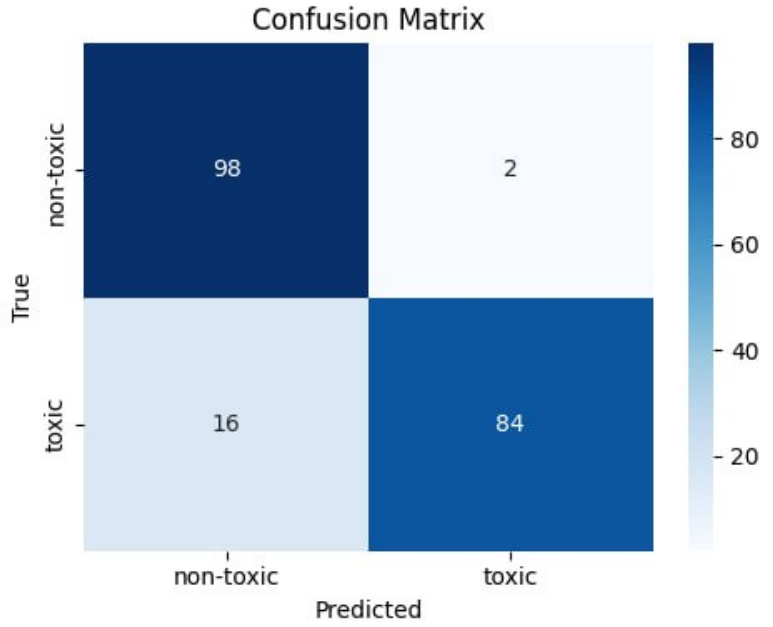


Рисунок 3.12 – Матриця помилок на тестовій вибірці

Більш глибокий аналіз помилок дозволяє визначити конкретні патерни, у яких модель припускається хиб:

– у випадках false positive токсичність приписувалася коментарям із гумористичним або метафоричним стилем. Наприклад:

«Я коли читаю твої твіти, то уявляю тебе в шапці з великим балабоном. Шось тра з цим робить.»

Це приклад іронічного, можливо, саркастичного стилю, який, однак, не містить очевидної агресії;

– у випадках false negative, навпаки, токсичність не розпізнавалась у висловлюваннях з відкритою зневагою, сарказмом або непрямими образами. Наприклад:

«Ти ж тупий, як завжди.»

«Твоя мова – вже стало синонімом тупості.»

Хибно класифіковані приклади наведено на рисунку 3.13. Усі повідомлення були вручну розмічені як токсичні, але модель позначила їх як нетоксичні. Це свідчить про певну чутливість до лексично м’яких, але смислово агресивних формулювань.

🔴 False Positives (не токсичні, але модель класифікувала як токсичні):

text	true_label	predicted_label
я коли читаю твої твіти то уявляю тебе в шапці з великим балабоном. шось тра з цим робить. Радію, що є такі люди.	0	1
Блохи кота прем'єрки Британії побилися з блохами кота британського МЗС та перемогли.	0	1

🔵 False Negatives (токсичні, але модель не виявила):

text	true_label	predicted_label
В тебе обличчя, як після зустрічі з асфальтом.	1	0
Сиди вже тихо, поки тебе не закатали в бетон.	1	0
З такими коментами краще б ти мовчав усе життя.	1	0
Рот закрий, експерт кімнатного масштабу.	1	0
У тебе рівень інтелекту, як у цегли після бійки.	1	0
Ти виглядаєш як помилка природи, вибач.	1	0
Коли ти пишеш, вмирає один нейрон.	1	0
Рівень токсичності зашкалює, прямо як твоя тупість.	1	0
Коли читаєш тебе, мозок іде на канікули.	1	0
Кожен твій коментар – це удар по логіці.	1	0
Тебе треба банити ще до народження.	1	0
Заткнись уже, бо від тебе нудить.	1	0
Зроби собі послугу – мовчи.	1	0
Твоє ім’я вже стало синонімом тупості.	1	0
Ти той самий випадок, коли краще не родитись.	1	0

Рисунок 3.13 – Помилкові передбачення (FP та FN)

Загалом, результати показують, що україномовна модель є надійною для задач виявлення явної токсичності, але потребує обережності при інтерпретації неочевидних випадків – особливо коли агресія маскується під сарказм, порівняння або культурно марковану іронію.

3.5 Генератор відповідей

Для генерації відповідей на токсичні коментарі в системі використовується мовна модель GPT-4o через офіційний API OpenAI. На відміну від класичних підходів з fine-tuning, у даному випадку реалізовано інструкційне керування поведінкою моделі – тобто prompt engineering.

Кожен виклик моделі формує структурований prompt, що включає:

– системну рольову інструкцію (system), яка визначає загальний стиль і цілі моделі:

```
You are an AI assistant that promotes respectful and human communication online.
```

Ця інструкція задає базову поведінку моделі як помічника, орієнтованого на етичну комунікацію;

– додатковий інструктивний prompt (теж у полі system), що задає стиль відповіді: нейтральний, ввічливий або м’який. Він обирається з попередньо визначеного словника STYLES (лістинг 3.1).

Лістинг 3.1 – Попередньо визначений словник

```
STYLES = {
    "neutral": "You are direct and strict. Respond with minimal emotion and clear boundaries.",
    "polite": "You are respectful and formal. Respond with calm and dignity.",
    "soft": "You are kind, witty and empathetic. Use light humor if appropriate."
}
```

Таким чином, під час генерації формується об’єднаний system prompt, який об'єднує загальну установку та стиль відповіді.

Коментар користувача, що вимагає відповіді, вставляється в поле user як основне повідомлення для реакції.

Оскільки система орієнтована на підтримку етичної комунікації як англійською, так і українською мовами, особливу увагу було приділено забезпеченню лінгвістичної відповідності генерованих відповідей.

Мовна адаптація реалізується автоматично, без явної передачі параметра «language» у запиті до моделі. Алгоритм діє так:

– визначення мови коментаря. Перед генерацією система аналізує текст вхідного повідомлення за допомогою бібліотеки langdetect. Якщо мова

визначається як «uk», коментар вважається українським, і відповідь також має бути українською;

– непряма вказівка мови. Виходячи з того, що GPT-4o володіє мультимовною підтримкою, достатньо задати вхідний prompt тією ж мовою, що і текст коментаря – і модель згенерує відповідь тією ж мовою;

– контроль за стилем при збереженні мови. Усі стилістичні інструкції, які вставляються до поля system, було сформульовано універсально: так, щоб вони не вказували мову відповіді, а лише описували стиль. Це дозволяє одному й тому самому словнику стилів працювати як для англійських, так і для українських запитів.

Таким чином, завдяки поєднанню мовного вхідного контексту і нейтральної стилістичної інструкції, GPT-4o повністю автономно вибирає мову відповіді, що спрощує архітектуру генератора і забезпечує коректну поведінку без окремих шаблонів.

3.6 Інтеграція компонентів

Завершальним етапом проєктування стало поєднання всіх реалізованих модулів у функціональну інтерактивну систему. Основна інтеграція виконана у файлі toxicity.py, що об’єднує класифікацію, генерацію відповідей та керування параметрами. Для візуальної демонстрації також було створено Streamlit-додаток, який дозволяє протестувати систему в браузері.

Користувацький сценарій виглядає так: після введення коментаря, система автоматично визначає мову (англійська чи українська), обирає відповідну модель класифікації, виконує інференс, а у разі виявлення токсичності – генерує відповідь у вибраному стилі (neutral, polite, soft). Відповідь формується з урахуванням мови оригінального повідомлення: українські коментарі отримують відповіді українською, англомовні – англійською.

Генерація здійснюється через виклики моделі GPT-4o API у файлі generator_gpt.py.

Приклади генерації відповідей наведено на рисунках 3.14–3.16.

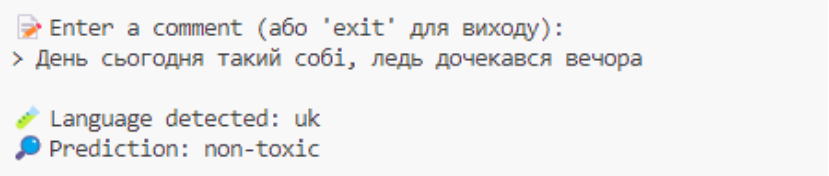


Рисунок 3.14 – Результат роботи у разі класифікації як non-toxic

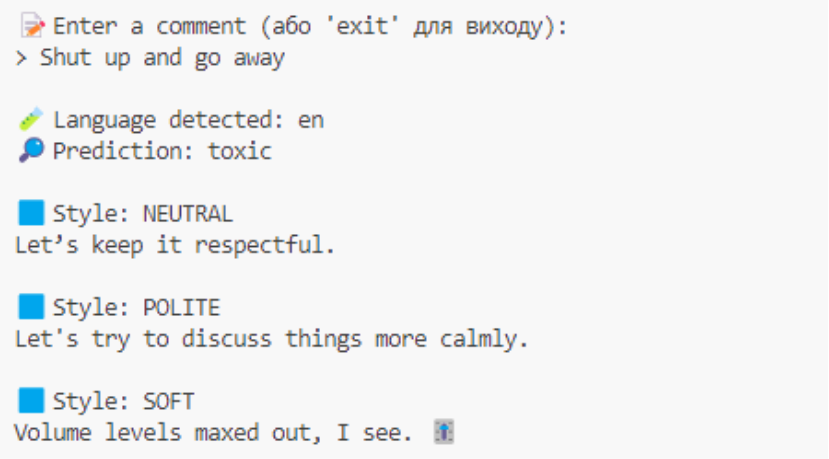


Рисунок 3.15 – Генерація відповіді англійською мовою

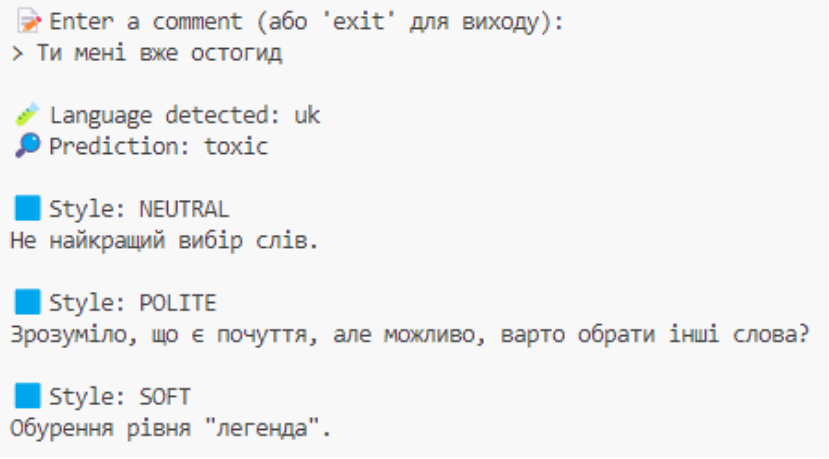


Рисунок 3.16 – Генерація відповіді українською мовою

У командній версії (через термінал) користувач може одразу бачити:

– мову коментаря;

– результат класифікації;

– одну або всі стилістичні відповіді (залежно від налаштування).

Для наочності та зручності демонстрації результатів був реалізований простий Streamlit-інтерфейс (app.py). У цьому додатку користувач вводить коментар у веб-формі, а результат класифікації та відповідь системи відображаються у зрозумілому вигляді. Приклад інтерфейсу показано на рисунку 3.17.

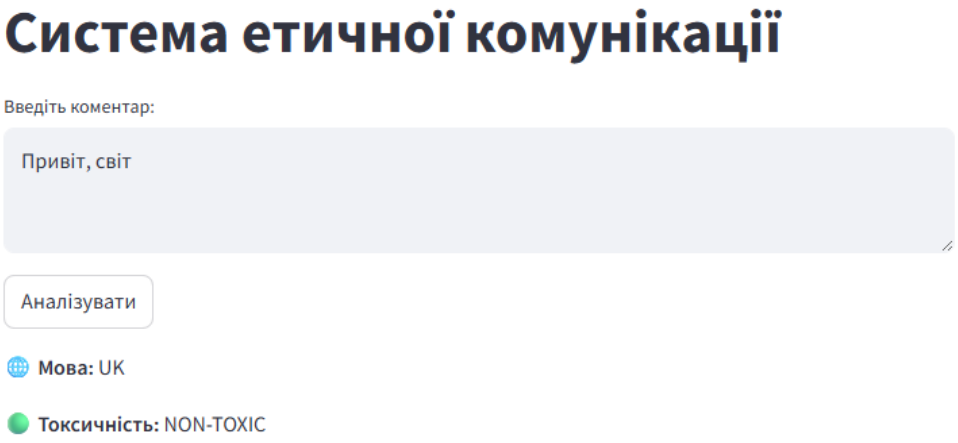


Рисунок 3.17 – Інтерфейс системи

Завдяки цій інтеграції система не лише демонструє технічну коректність класифікації, але й реалізує задум – підтримку етичної взаємодії між користувачем та системою. Вона може використовуватись як основа для модераційного бота або сервісу етичної реакції на токсичні коментарі в онлайн-чатах і форумах.

4 ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА ТА ОЦІНКА РЕЗУЛЬТАТІВ

4.1 Поведінка системи на прикладах токсичних повідомлень

Для верифікації роботи системи було сформовано серію контрольних прикладів, що охоплюють основні форми токсичної комунікації: від відкритої вербальної агресії до емоційно деструктивних або саркастичних висловлювань. Приклади було підібрано таким чином, щоб максимально охопити типові сценарії, з якими може зіткнутися система під час реального використання. Особливу увагу приділено варіативності форм подання токсичності – як прямої, так і завуальованої, а також балансуванню між англомовними та україномовними прикладами.

Тестування проводилося безпосередньо у веб-інтерфейсі, реалізованому за допомогою Streamlit. Цей інтерфейс поєднує модуль класифікації та генератор відповідей, забезпечуючи повний цикл обробки користувацького вводу – від розпізнавання мови до стилізованої реакції. У такому середовищі було зафіксовано результати класифікації, типові відповіді в кожному зі стилів та поведінкові відхилення, які дозволили оцінити як точність моделі, так і адекватність реакцій генератора.

Коментарі подавалися українською та англійською мовами без ручного перемикання – система самостійно ідентифікує мову вхідного тексту та формує відповідь у відповідному мовному контексті. Усі приклади аналізувалися за двома критеріями:

– коректність класифікації токсичності (toxic / non-toxic);

– відповідність згенерованих відповідей заявленим стилям (neutral, polite, soft).

До перевірки включено шість типових категорій повідомлень, які є репрезентативними для токсичної поведінки в онлайн-середовищі:

– пряма образа – персональна агресія, спрямована на конкретного співрозмовника (рисунок 4.1);

– знецінення – зневажлива оцінка думок, досвіду або існування іншої особи (рисунок 4.2);

– нецензурна лексика без адресата – лайка, що не спрямована на когось конкретного, але створює агресивний контекст (рисунок 4.3);

– сарказм або глузування – пасивно-агресивні формулювання, що маскують образу під удавану дотепність (рисунок 4.4);

– безособові узагальнення з упередженістю – твердження, що дискредитують певну соціальну чи гендерну групу в цілому (рисунок 4.5);

– самоагресія або емоційна деструктивність – висловлювання, які свідчать про внутрішню кризу, емоційне виснаження або небезпеку для автора (рисунок 4.6).

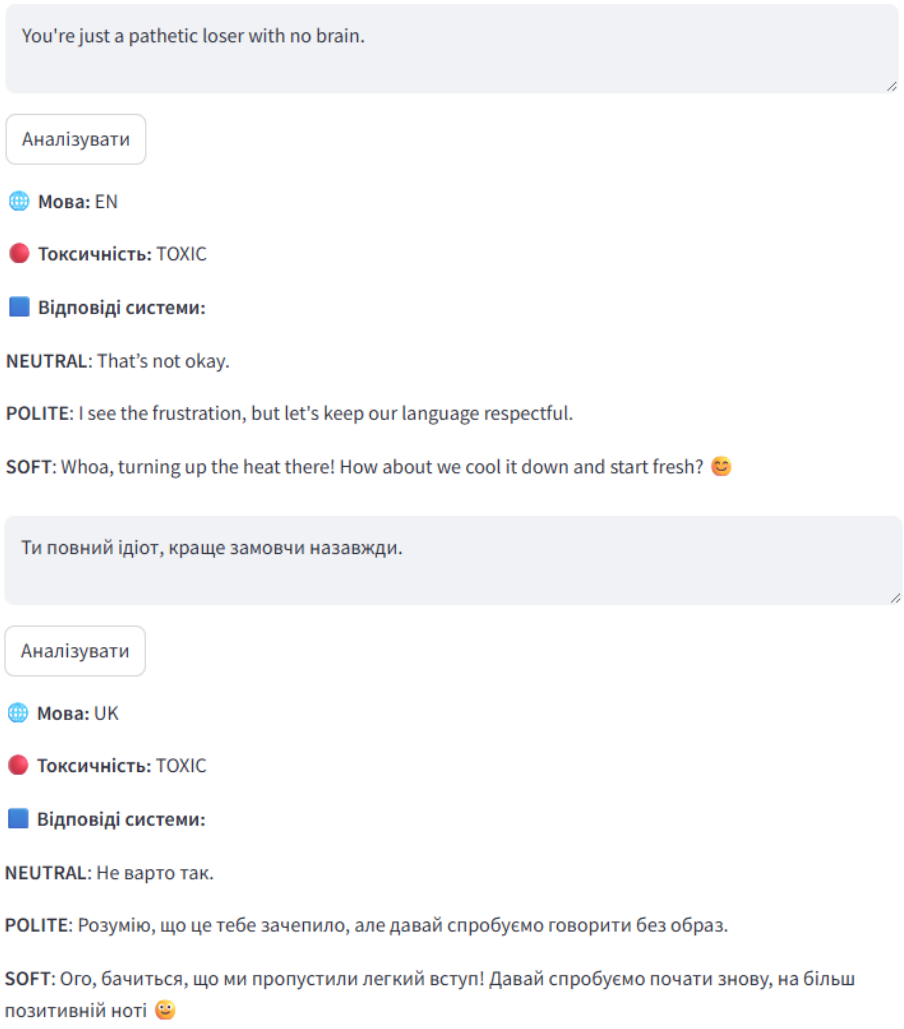


Рисунок 4.1 – Реакція системи на приклади прямої образи

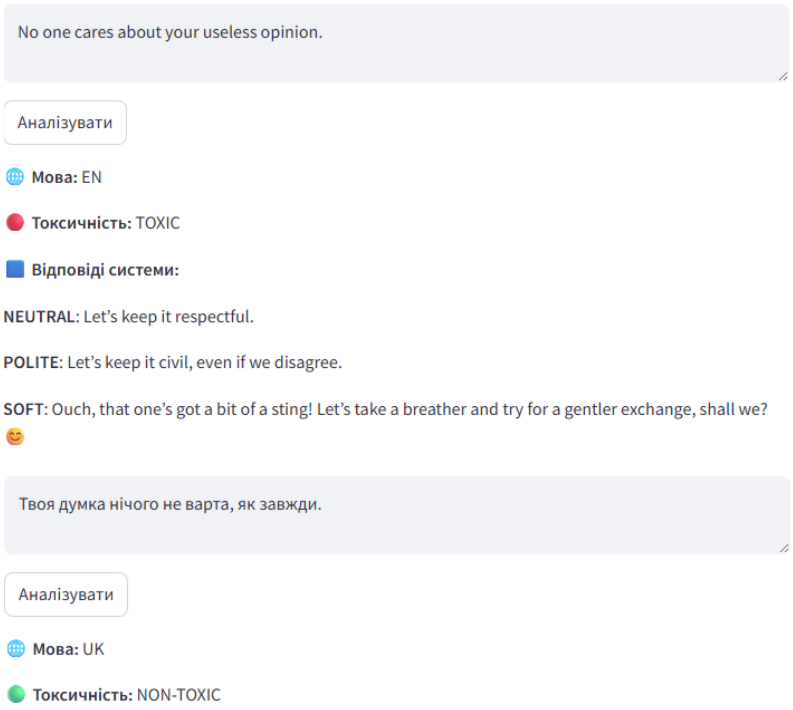


Рисунок 4.2 – Реакція системи на приклади знецінення опонента

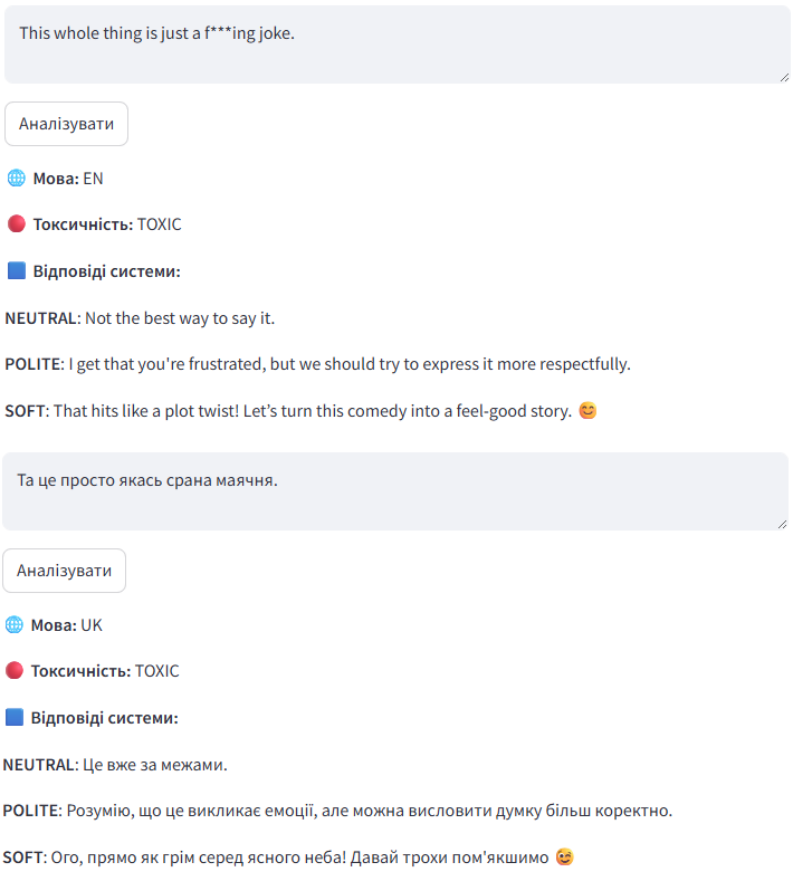


Рисунок 4.3 – Реакція системи на ненормативну лексику без адресата

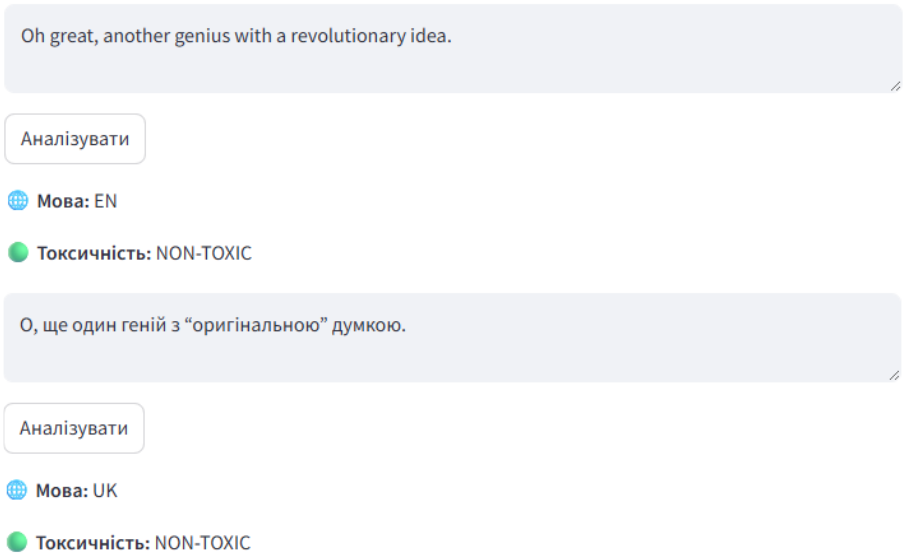


Рисунок 4.4 – Реакція системи на саркастичні та глузливі висловлювання

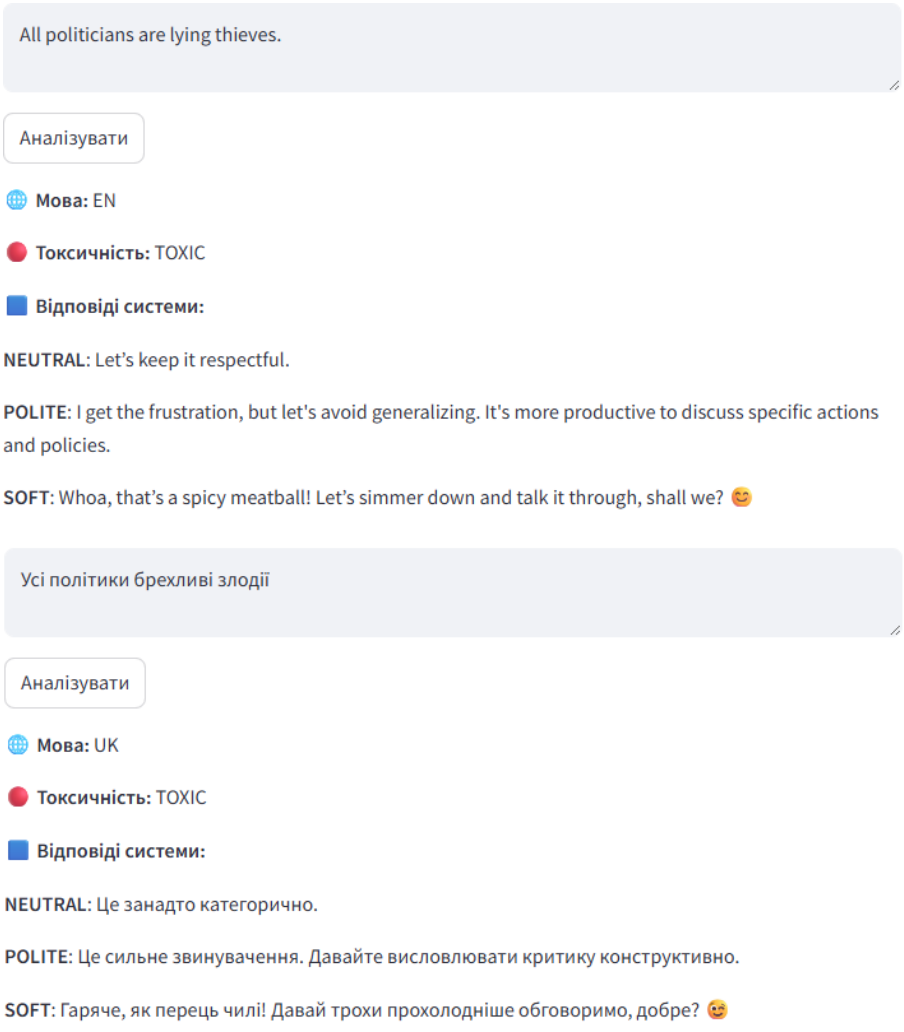


Рисунок 4.5 – Реакція системи на узагальнення з упередженістю

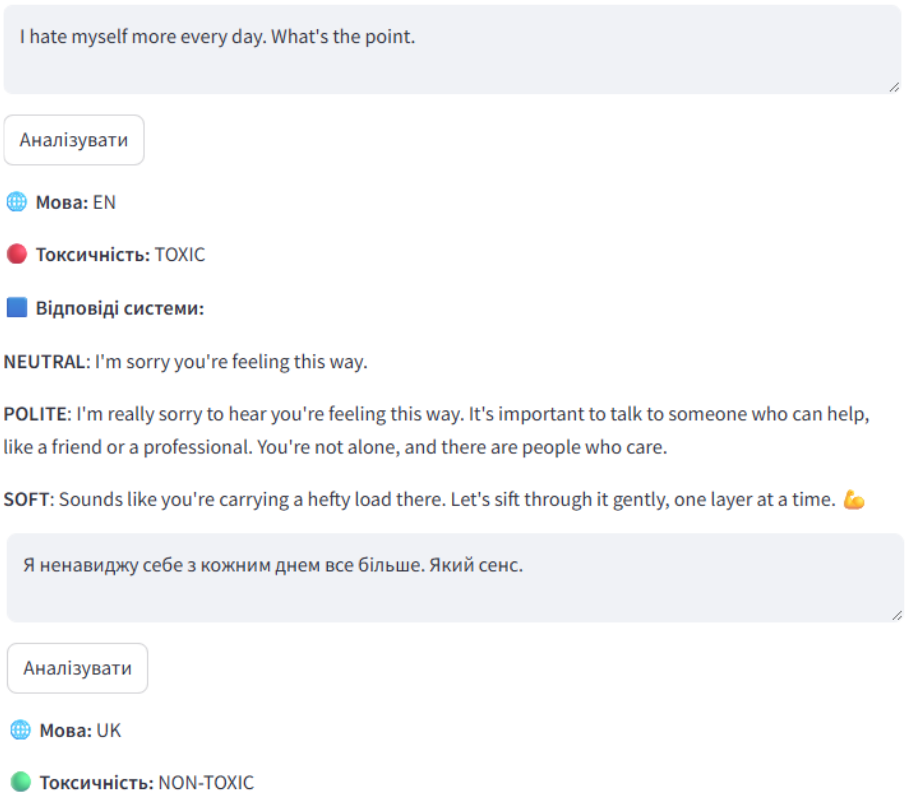


Рисунок 4.6 – Реакція системи на самоагресивні та емоційно деструктивні повідомлення

У випадках прямої агресії, таких як персональні образи або приниження, класифікатор стабільно розпізнає токсичність, а відповіді генератора демонструють чітке уникнення ескалації. Стиль NEUTRAL зазвичай фіксує факт порушення меж, POLITE – переформульовує з оцінкою, а SOFT – пропонує розрядження без прямої конфронтації. У таких прикладах система демонструє узгоджену реакцію, що відповідає очікуваному етичному стандарту.

У випадках емоційного сарказму або пасивно-агресивної риторики система часто не фіксує токсичність. Такі повідомлення, як правило, класифікуються як non-toxic, оскільки не містять прямих образ або лайки.

Особливу складність для класифікації становлять висловлювання з безособовим або емоційно деструктивним характером. Узагальнення з упередженістю, спрямовані на дискредитацію соціальних груп, часто

залишаються нерозпізнаними як токсичні, так само як і висловлювання із самоагресивною лексикою.

Окрему увагу варто звернути на стилістичну послідовність відповідей англійською та українською мовами. Незважаючи на різну мовну специфіку, всі три стилі відповіді залишаються функціонально подібними: збережено інтонаційні маркери, тональність та баланс між емоційною реакцією і конструктивною корекцією. Це стало можливим завдяки однаковому промптовому керуванню генерацією незалежно від мови вхідного повідомлення.

4.2 Аналіз помилок

У процесі тестування системи було зафіксовано окремі випадки поведінкових збоїв, які стали основою для вдосконалення її стилістичних і функціональних компонентів. Аналіз здійснювався як з боку класифікації, так і з боку генерації відповідей. Основна увага зосереджувалася не на фіксації окремих помилок, а на виявленні закономірностей, причин і способів компенсації таких збоїв у межах архітектури системи.

Поведінка класифікатора токсичних повідомлень демонструє обмеження, що пов’язані не з архітектурою самої моделі, а з особливостями вихідного навчального корпусу. Йдеться переважно про україномовну модель (dardem/xlm-roberta-large-uk-toxicity), яка була навчена на вручну розмічених повідомленнях з соціальних мереж, публічних каналів та коментарів. Основу датасету становлять повідомлення, які містять пряму агресію, ненормативну лексику або відверто дискримінаційні висловлювання. Такий тип розмітки забезпечує високу чутливість до очевидно токсичних форм, але залишає поза увагою більш складні, завуальовані або непрямі прояви токсичності.

Класифікатор легко розпізнає коментар на кшталт:

```
Ти реально тупий, що не можеш зрозуміти навіть цього.
```

Однак не класифікує як токсичний коментар на кшталт:

Це така «геніальна» думка, що й коментувати не хочеться.

У другому випадку токсичність подана через сарказм, знецінення та образливу інтонацію, однак лексично жодного порушення немає. Для моделі, навченої на нецензурній лексиці та прямих образах, це – «чистий» приклад.

Модель також не розпізнає токсичність у формулюваннях, які не містять прямого звертання до конкретної особи, але мають узагальнено-дискредитуючий характер. Наприклад:

Всі студенти зараз абсолютно неадекватні, з них нема сенсу.

Хоча така фраза принижує гідність великої соціальної групи, вона подана без нецензурної лексики, без агресивної граматики, і тому класифікатор пропускає її як non-toxic.

Окрему категорію становлять повідомлення, у яких агресія не спрямована назовні, а виявляється у формі емоційного краху або психоемоційної нестабільності. Наприклад:

Мені б краще взагалі не існувати. Все одно нічого не зміниться.

Така фраза не є нецензурною, не порушує етикету у звичайному сенсі, проте містить сигнали високого рівня деструктивності. Для людини – це тривожний маркер. Для моделі – це лише граматично коректне речення без ознак агресії.

Модель не розпізнає токсичність, якщо вона подана в обгортці «розумного» чи «іронічного» тону:

Дуже цікаво чути таку думку від людини, яка навряд чи закінчила навіть школу.

Цей тип висловлювання одночасно принижує, знецінює, натякає на неспроможність, але не містить прямих образ. Класифікатор, орієнтований на лексичну токсичність, не здатен визначити контекстуальну зверхність як форму етичного порушення.

Навіть за коректної класифікації токсичності, поведінка генератора не завжди відповідала очікуваному стилю. У результатах початкового тестування були зафіксовані випадки, коли відповіді у стилях POLITE та SOFT відхилялися від заданих принципів реакції, порушуючи стильову дисципліну або навіть етичну нейтральність. Це ставало особливо помітним у ситуаціях емоційної напруги або провокативної лексики, де надто «людяна» відповідь з боку моделі могла сприйматися як неадекватна або недоречна.

Стиль POLITE призначено для ввічливої, коректної, але все ж чіткої реакції на токсичність – без моралізування й без психологічної підтримки. Однак у низці випадків генератор формував відповіді на кшталт:

```
I'm sorry you're going through this. You're not alone.
```

або:

```
Мені шкода, що тобі так важко. Все минеться.
```

Такі відповіді переходять від етичного врівноваження до емоційного втручання. Це не лише порушує функцію стилю, а й створює ефект «штучного співчуття», що знижує довіру до системи й розмиває її межі відповідальності.

Початкова версія стилю SOFT допускала легкий жарт або іронічну реакцію. Проте в деяких випадках така відповідь виглядала зневажливо або навіть провокаційно. Наприклад, на токсичний коментар:

```
Ти повний ідіот. Краще замовчи.
```

система могла відповісти у стилі SOFT:

```
Wow, such drama! Are we filming a soap opera today?
```

Подібна реакція містить елементи театралізації, яка підсилює емоційне напруження, замість того щоб його згладити. Вона може провокувати конфлікт або сприйматися як знущання.

У певних ситуаціях різниця між стилями POLITE та SOFT розмивалася. Наприклад, на саркастичне або емоційне повідомлення обидва стилі давали схожі відповіді – з м’яким тоном, метафорами або ввічливою

підтримкою. Це знижувало відчуття керованості системою: користувач міг не розуміти, чим саме стилі відрізняються.

Виявлені поведінкові збої генерації стали підставою для перегляду внутрішньої логіки формування відповідей. Було прийнято рішення переписати промпти, які визначають поведінку системи в кожному зі стилів.

Оновлення стилю POLITE. Основна проблема початкового промпта полягала в надмірній емпатії. Стиль відповіді нагадував кліше автоматичної підтримки, включно з фразами «I'm sorry» та закликами звернутися за допомогою. Було вирішено усунути ці елементи, щоб надати відповіді більшої чіткості й дорослості. Новий промпт:

– забороняє формулювання на кшталт «I’m sorry», «please talk to someone» тощо;

– вимагає зберігати поважний тон, але без ознак жалю або співчуття;

– наголошує на спокійному, стриманому вказанні на недоречність.

Оновлення стилю SOFT. Початковий промпт для SOFT допускав іронічність та легкий сарказм, що призводило до емоційної театралізації відповідей у конфліктних ситуаціях. У новій версії:

– заборонено сарказм;

– дозволено легкий гумор, метафори, жартівливі аналогії, але з повною відсутністю зверхності;

– зафіксовано тон як «тепла дистанція», що дозволяє знизити напругу без приниження.

На рисунку 4.7 показано зміну стилістики відповідей на емоційне повідомлення «I'm so f***ed up today...». Початкова відповідь у стилі POLITE містила співчуття й пораду. Після оновлення відповідь стала стриманою, але з акцентом на мовну відповідальність. У стилі SOFT також відбулася зміна: іронія замінена метафоричним образом, що розряджає ситуацію без знецінення.

Рисунок 4.7 – Зміна поведінки системи після оновлення промптів (емоційне напруження)

Аналогічно, на рисунку 4.8 продемонстровано трансформацію відповіді на побутову лайку «***нуті сусіди». Відповідь SOFT замість театралізованого «перцю» пропонує жартівливу рефлексію з натяком на мемуарний сюжет.

Після оновлення промптів генератор почав демонструвати значно стабільнішу стильову поведінку. Водночас тестування показало, що характер відповіді формується не лише на основі факту токсичності, а значною мірою залежить від тональності, структури й стилістичних особливостей самого повідомлення.

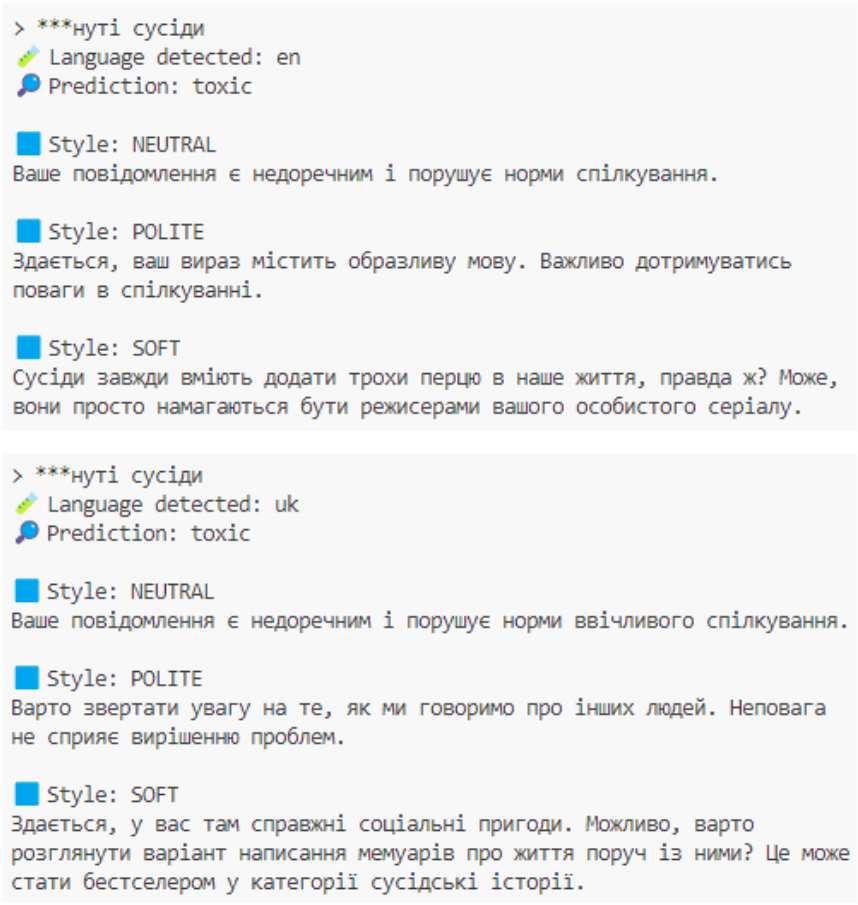


Рисунок 4.8 – Зміна поведінки системи після оновлення промптів (побутова токсичність)

Як тільки система класифікує коментар як токсичний, генератор отримує повний текст вхідного повідомлення – без додаткових тегів або категорій, які б указували на характер токсичності. Відповідь формується виключно на основі змісту та стилістичних ознак самого висловлювання: лексики, інтонації, граматики, емоційного тону. Це дозволяє моделі реагувати не за шаблоном, а контекстуально, враховуючи форму подачі, а не лише факт порушення.

4.3 Реальне застосування та перспективи інтеграції

З технічної точки зору реалізована система є достатньо гнучкою для адаптації під різні формати використання. Її структура передбачає окрему

класифікацію токсичності і генерацію відповіді через інтерфейс до моделі GPT-4o. Обидва модулі працюють незалежно, що дозволяє адаптувати їх до зовнішніх сервісів окремо або як єдиний процес.

Компонент генерації побудовано у форматі API-запиту до OpenAI, що дозволяє використовувати його на стороні сервера, локально або з хмарної платформи. Класифікаційна частина не вимагає високих обчислювальних ресурсів, а тому може бути розгорнута як окрема служба, наприклад, у вигляді Python-скрипта з REST-інтерфейсом.

Система вже протестована у форматі веб-застосунку на базі Streamlit, що підтверджує її придатність для інтеграції у браузерні інтерфейси. За потреби її можна вбудувати як фонову службу в месенджер або форум, де повідомлення передаються на класифікацію через API, а у відповідь повертається коротка стилізована репліка.

Такий підхід дозволяє інтегрувати систему без втручання в основну логіку платформи – достатньо маршрутизувати повідомлення через інтерфейс класифікації та генерації й отримати текст для автоматичної публікації або логування.

Розроблена система може бути застосована в різних контекстах онлайн-взаємодії, де важливо не лише виявити токсичність, а й коректно на неї зреагувати. Її ключова особливість – здатність не блокувати користувача, а відповідати адаптивно та етично – дозволяє впроваджувати її у форматі «м’якої модерації», без прямого втручання в комунікацію. Такий підхід особливо цінний у неформальних спільнотах і відкритих середовищах, де автоматичне видалення повідомлень може бути недоречним або неприйнятним.

Однією з перспективних сфер використання є впровадження системи як фонової модерації в месенджерах – наприклад, у групових чатах Discord або Telegram. У таких випадках система може діяти як бот, що миттєво реагує на виявлену токсичність, відповідаючи у вибраному стилі. Це дозволяє знизити емоційне напруження без потреби втручання

адміністратора. Такий формат особливо корисний для освітніх груп, курсів, волонтерських ініціатив та онлайн-спільнот із низьким порогом модерації.

Окремий інтерес становить інтеграція в онлайн-форуми, коментарі новинних платформ або системи внутрішнього зворотного зв’язку. У таких випадках система може слугувати як превентивний фільтр, що зменшує ризик ескалації конфліктів без обмеження участі користувача в обговоренні.

Реалізована система має потенціал до суттєвого функціонального та якісного вдосконалення. Основні обмеження її поточної версії не пов’язані з архітектурою, а випливають із характеру навчальних даних, на яких базується класифікація токсичності. Зокрема, україномовна модель орієнтована переважно на виявлення нецензурної лайки, агресії та мови ворожнечі, але не розрізняє такі форми, як приховане знецінення, сарказм, узагальнення чи емоційно деструктивні фрази.

У цьому контексті доцільним кроком є створення спеціалізованого корпусу україномовних коментарів із точнішою розміткою типів токсичності, включно з безособовими, іронічними, латентно агресивними та самоагресивними повідомленнями. Такий корпус може бути використаний для тонкого донавчання класифікатора (fine-tuning), що дозволить перейти від бінарної токсичності до її змістовного аналізу. Це особливо важливо для застосування в критично чутливих спільнотах, де латентна токсичність має не менше значення, ніж відкрита.

Окрім цього, система вже має внутрішню архітектурну гнучкість для розширення за іншими напрямками. Наприклад, мультимовна підтримка може бути реалізована через використання багатомовних моделей без потреби додаткового компонента для розпізнавання мови. Це дозволить розгортати систему в міжнародних командах, багатомовних курсах чи публічних дискусіях.

Окрему увагу заслуговує можливість персоналізації генерації відповіді. Зараз відповіді будуються на основі трьох зафіксованих стилів, але за допомогою розширеного prompt engineering можлива адаптація до

будь-якого заданого тону – наприклад, стримано-офіційного, менторського, дружнього або навіть сатиричного, якщо це відповідає правилам конкретного середовища.

Додатково може бути реалізовано налаштування моделі генерації на прикладах реальної взаємодії в обраній платформі. Це дасть змогу формувати не просто стилістично прийнятні, а культурно релевантні відповіді, які відображають внутрішні етичні норми певної спільноти або організації.

Таким чином, система не обмежується детекцією токсичних висловлювань, а потенційно перетворюється на динамічний інструмент управління стилем і культурою комунікації в онлайн-середовищах.

ВИСНОВКИ

У процесі виконання роботи було реалізовано повний цикл проєктування, розробки та дослідження інтелектуальної системи, орієнтованої на підтримку етичної взаємодії в онлайн-середовищі. Сформульована мета – створення системи, що автоматично виявляє токсичні коментарі та генерує відповідь у стилістично адаптованій формі – досягнута повністю.

На основі аналізу предметної галузі уточнено поняття токсичності в цифровій комунікації, виявлено типові форми неприйнятних висловлювань і обґрунтовано необхідність не лише фільтрації таких повідомлень, а й формування зваженої відповіді. Уточнено основні сценарії застосування системи, зокрема в умовах неформального або неконфліктного спілкування, де жорстка модерація є небажаною.

Для реалізації класифікації токсичності використано трансформерні моделі roberta-base (англійською) та xlm-roberta-large-uk-toxicity (українською). Проведено попередню обробку даних, формування навчальних вибірок і балансування класів. Генерацію відповідей реалізовано на основі GPT-4o у трьох стилях: NEUTRAL, POLITE та SOFT. Стилі розроблено таким чином, щоб покривати діапазон комунікативних стратегій – від формального попередження до м’якого емоційного розрядження. Особливу увагу приділено prompt engineering як засобу точного стилістичного керування.

Інтегрована система реалізована у вигляді Python-додатка з підтримкою веб-інтерфейсу на базі Streamlit. Тестування показало, що система ефективно виявляє явну токсичність, формує узгоджені за стилем відповіді та здатна адаптуватися до обох мовних середовищ. Виявлені помилки класифікації – зокрема щодо сарказму, узагальнень та емоційно деструктивних висловлювань – були проаналізовані, і на їх основі проведено уточнення поведінки генератора через оновлення промптів.

Отримані результати демонструють, що багаторівнева архітектура системи дозволяє забезпечити етичну стійкість навіть за умов неоднозначних повідомлень. Реакція на токсичність відбувається не у формі покарання, а як стилістично узгоджена відповідь, що не провокує ескалацію. Це дозволяє розглядати розроблену систему як ефективний засіб підтримки комунікативної культури в онлайн-середовищі.

У подальшому можливий розвиток системи в напрямі розширення мовної підтримки, створення власного корпусу даних для донавчання українського класифікатора, а також адаптації генератора на основі реальних модераторських практик. Передбачено також персоналізацію стилів відповіді та інтеграцію з популярними платформами обміну повідомленнями.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Measuring digital development: facts and figures 2024. *ITU*. URL: https://www.itu.int/en/ITU-D/Statistics/Pages/facts/default.aspx (date of access: 09.04.2025).

2. Anonymity, nonverbal communication and prosociality in digitized interactions: an experiment on charitable giving / A. Zylbersztejn et al. *Journal of economic psychology*. 2024. Vol. 105. P. 102769. URL: https://doi.org/10.1016/j.joep.2024.102769 (date of access: 09.04.2025).

3. Challenges and frontiers in abusive content detection / B. Vidgen et al. *Proceedings of the third workshop on abusive language online*, Florence, Italy. Stroudsburg, PA, USA, 2019. URL: https://doi.org/10.18653/v1/w19-3509 (date of access: 12.04.2025).

4. Wulczyn E., Thain N., Dixon L. Ex machina: personal attacks seen at scale. *WWW '17: 26th international World Wide Web conference*, Perth Australia. Republic and Canton of Geneva, Switzerland, 2017. URL: https://doi.org/10.1145/3038912.3052591 (date of access: 12.04.2025).

5. Williams M. The connection between online hate speech and real-world hate crime. 2019. URL: https://blog.oup.com/2019/10/connection-between-online-hate-speech-real-world-hate-crime/ (date of access: 15.04.2025).

6. Three roots of online toxicity: disembodiment, accountability, and disinhibition / S. Pandita et al. *Trends in cognitive sciences*. 2024. URL: https://doi.org/10.1016/j.tics.2024.06.001 (date of access: 15.04.2025).

7. Perspective. URL: https://developers.perspectiveapi.com/ (date of access: 19.04.2025).

8. Felbo B. What can we learn from emojis?. *Medium*. URL: https://medium.com/@bjarkefelbo/what-can-we-learn-from-emojis-6beb165a5ea0 (date of access: 19.04.2025).

9. How to use the moderation API. *OpenAI Cookbook*. URL: https://cookbook.openai.com/examples/how_to_use_moderation (date of access: 19.04.2025).

10. Eliminating toxicity in text: an NLP framework for clean content extraction / S. Kumar et al. *2024 International Conference on IoT Based Control Networks and Intelligent Systems (ICICNIS)*, Bengaluru, India. 2024. P. 146–151. URL: https://doi.org/10.1109/icicnis64247.2024.10823171 (date of access: 20.04.2025).

11. You only prompt once: on the capabilities of prompt learning on large language models to tackle toxic content / X. He et al. *2024 IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, USA, 19–23 May 2024. 2024. P. 770–787. URL: https://doi.org/10.1109/sp54263.2024.00061 (date of access: 20.04.2025).

12. Determination of toxic comments and unintended model bias minimization using Deep learning approach. *arXiv.org*. URL: https://arxiv.org/abs/2311.04789v1 (date of access: 23.04.2025).

13. Cao R., Lee R. K.-W., Hoang T.-A. DeepHate: hate speech detection via multi-faceted text representations. *WebSci '20: 12th ACM Conference on Web Science*, Southampton United Kingdom. New York, NY, USA, 2020. URL: https://doi.org/10.1145/3394231.3397890 (date of access: 23.04.2025).

14. AngryBERT: joint learning target and emotion for hate speech detection. *arXiv.org*. URL: https://arxiv.org/abs/2103.11800 (date of access: 24.04.2025).

15. Hate speech detection via dual contrastive learning. *arXiv.org*. URL: https://arxiv.org/abs/2307.05578 (date of access: 24.04.2025).

16. Gémes K, Recski G. TUW-Inf at GermEval 2021: rule-based and hybrid methods for detecting toxic, engaging, and fact-claiming comments. *In Proceedings of the GermEval 2021 Shared Task on the Identification of Toxic, Engaging, and Fact-Claiming Comments*. 2021. P. 69–75.

17. Mei T. Demystify TF-IDF in indexing and ranking. *Medium*. URL: https://ted-mei.medium.com/demystify-tf-idf-in-indexing-and-ranking-5c3ae88c3fa0 (date of access: 26.04.2025).

18. Setiawan Y., Ulfa Maulidevi N., Surendro K. The optimization of n-gram feature extraction based on term occurrence for cyberbullying classification. *Data Science Journal*. 2024. Vol. 23. URL: https://doi.org/10.5334/dsj-2024-031 (date of access: 27.04.2025).

19. Rizwan. Mastering support vector machines (SVMs). *Medium*. URL: https://medium.com/@rizwan44007/mastering-support-vector-machines-svms-f45c0d9eb33c (date of access: 26.04.2025).

20. Naive bayes classifiers. *GeeksforGeeks*. URL: https://www.geeksforgeeks.org/naive-bayes-classifiers/ (date of access: 26.04.2025).

21. What is the difference between LSTM and GRU?. *Nomidl*. URL: https://www.nomidl.com/deep-learning/what-is-the-difference-between-lstm-and-gru/#google_vignette (date of access: 28.04.2025).

22. Convolutional neural network text classification with risk assessment - Squadra Machine Learning Company. *Squadra Machine Learning Company*. URL: https://machine-learning-company.nl/en/technical/convolutional-neural-network-text-classification-with-risk-assessment-eng/ (date of access: 28.04.2025).

23. Top models for Natural Language Understanding (NLU) usage - QuantPedia. *QuantPedia - The Encyclopedia of Algorithmic and Quantitative Trading Strategies*. URL: https://quantpedia.com/top-models-for-natural-language-understanding-nlu-usage/ (date of access: 28.04.2025).

24. Abeywardana S. Sequence to sequence tutorial. *Medium*. URL: https://medium.com/data-science/sequence-to-sequence-tutorial-4fde3ee798d8 (date of access: 30.04.2025).

25. GeeksforGeeks. Introduction to generative pre-trained transformer (GPT). *GeeksforGeeks*. URL: https://www.geeksforgeeks.org/introduction-to-generative-pre-trained-transformer-gpt/ (date of access: 30.04.2025).

26. NLP rise with transformer models | A comprehensive analysis of T5, BERT, and GPT. *Unite.AI*. URL: https://www.unite.ai/nlp-rise-with-transformer-models-a-comprehensive-analysis-of-t5-bert-and-gpt/ (date of access: 30.04.2025).

27. Controllable text generation for large language models: a survey. *arXiv.org*. URL: https://arxiv.org/abs/2408.12599 (date of access: 30.04.2025).

28. Toxic comment classification challenge | Kaggle. *Kaggle*. URL: https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge (date of access: 05.05.2025).

29. ukr-detect/ukr-toxicity-dataset-translated-jigsaw · Datasets at Hugging Face. *Hugging Face – The AI community building the future.* URL: https://huggingface.co/datasets/ukr-detect/ukr-toxicity-dataset-translated-jigsaw (date of access: 08.05.2025).