

КИЇВСЬКИЙ СТОЛИЧНИЙ УНІВЕРСИТЕТ ІМЕНІ БОРИСА ГРІНЧЕНКА
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА МАТЕМАТИКИ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

«Допущено до захисту»

Завідувач кафедри
к.т.н., доцент кафедри
комп'ютерних наук

_____Ірина МАШКІНА

«___»_____2025р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня «бакалавр»
Спеціальність 122 «Комп'ютерні науки»
Освітня програма 122.00.01 Інформатика

Тема:

**Використання методів машинного навчання та штучного інтелекту для
розробки новинного бота**

Виконав:

студент групи ІНб-2-214.0д
Кривошесєв Іван Сергійович

(підпис)

Науковий керівник:

к.т.н доцент кафедри комп'ютерних наук
_____Носенко Т.І.

Київ – 2025

Київський столичний університет імені Бориса Грінченка
Факультет інформаційних технологій та математики
Кафедра комп'ютерних наук

«Затверджую»

Завідувач кафедри
к.т.н., доцент кафедри
комп'ютерних наук

Ірина МАШКІНА

«___» _____ 2025р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

**«Використання методів машинного навчання та штучного інтелекту для
розробки новинного бота»**

Виконавець – студент групи ІНб-2-21-4.0д

Кривошесів Іван Сергійович

Вихідні дані: Тематичні українські та зарубіжні освітні ресурси. Наукова, навчальна та періодична та методична література за темою роботи.

Основні завдання: Провести аналіз існуючих методів машинного навчання та їх

застосування у сфері автоматизованого оброблення новин. Дослідити сучасні технології та бібліотеки, що використовуються для створення новинних ботів. Розробити модель машинного навчання для аналізу та класифікації новинного контенту. Реалізувати програмну систему новинного бота з використанням методів машинного навчання. Провести тестування та оцінити ефективність роботи новинного бота.

Пояснювальна записка: обсяг – 33 сторінки формату А4 комп'ютерного набору з дотриманням вимог стандарту і методичних рекомендацій кафедри

Графічні матеріали: скріншоти інтерфейсу чат-бота, коду.

Додатки:

Додаток А Б, В, Г, Д

Строк подання роботи на кафедру: «05.06.2025»

Календарний план

Дата	Індивідуальний календарний план
07.09.2024	Формування теми дипломної роботи та узгодження з науковим керівником
08.09.2024-28.09.2024	Аналіз поставленого завдання та існуючих рішень в сфері машинного навчання та новинних ботів
29.09.2024-29.10.2024	Визначення об'єкта та предмета дослідження, постановка задачі. Аналіз популярних алгоритмів машинного навчання.
30.10.2024-30.11.2024	Вибір інструментів для реалізації новинного бота. Розробка загальної концепції та архітектури
01.12.2024-31.12.2024	Визначення метрик ефективності роботи новинного бота. Підготовка проміжного звіту.
01.01.2025-28.01.2025	Розробка та імплементація моделі машинного навчання для аналізу і класифікації новин. Попереднє тестування моделі.
29.01.2025-22.02.2025	Реалізація програмного забезпечення новинного бота.
23.02.2025-26.03.2025	Проведення тестування на реальних даних. Аналіз отриманих результатів, коригування алгоритмів.
27.03.2025-29.04.2025	Оформлення дипломної роботи, написання висновків та рекомендацій. Підготовка до захисту: перевірка відповідності вимогам ВНЗ.
30.04.2025-30.05.2025	Фінальне редагування та підготовка презентації для захисту. Захист дипломної роботи.

РЕФЕРАТ

Кваліфікаційна робота: 43с., 13 рис., 18 посилань

Актуальність: У розвинених країнах – використовують розвинені технології, чат бот – одна з таких технологій. Чат боти є повсюди, починаючи з банківської справи, закінчуючи сферою розваг. Вони, чат боти, допомагають в вирішенні повсякденних питань, будь то питання з приводу комунальних послуг (чат бот “Дії”), або різноманітних питань (OpenAI\ChatGPT).

Тема: Використання методів машинного навчання та штучного інтелекту для розробки новинного бота

Об’єкт дослідження: розробка новинного чат боту.

Предмет дослідження: Методи машинного навчання та штучного інтелекту для розробки новинного бота

Мета роботи: Метою роботи є створення новинного бота, за допомогою мови програмування Python. Додавання базових функцій такого бота.

Завдання:

- Провести аналіз існуючих методів машинного навчання та їх застосування у сфері автоматизованого оброблення новин.
- Дослідити сучасні технології та бібліотеки, що використовуються для створення новинних ботів.
- Розробити модель машинного навчання для аналізу та класифікації новинного контенту.
- Реалізувати програмну систему новинного бота з використанням методів машинного навчання.
- Провести тестування та оцінити ефективність роботи новинного бота.
- Зробити висновки щодо перспектив розвитку новинних ботів та можливих покращень моделі.

Основні результати: розроблення чат боту

Ключові слова: програмування; бот; Python; ШІ .

ЗМІСТ

ВСТУП	5
Розділ I. Новинні боти та машинне навчання: огляд та аналіз	7
1.1 Новинні боти: типи, функції, сфери застосування	7
1.2 Машинне навчання та штучний інтелект в обробці новин.	8
1.3 Методи класифікації текстів та обробки природної мови (NLP)	9
1.4. Огляд існуючих новинних ботів з використанням ШІ	10
Висновки до розділу 1	14
Розділ II. Розробка новинного боту	15
2.1 Вибір та обґрунтування методів машинного навчання	15
2.2 Проектування архітектури новинного бота	17
2.3 Розробка модуля збору новин.	20
2.4 Реалізація модуля класифікація новин	23
2.5 Створення модуля генерації новини	24
2.6 Розробка інтерфейсу користувача	26
Висновки до розділу 2	29
Розділ III. Тестування та оцінка ефективності	30
3.1 Планування та проведення тестування	30
3.2 Оцінка точності класифікації новини	31
3.3. Оцінка якості генерації відповідей	32
3.4. Аналіз результатів тестування та формулювання висновків	33
Висновки до розділу 3	34
ВИСНОВКИ	35
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	36

ВСТУП

У сучасному інформаційному суспільстві швидкість отримання та обробки новин є критично важливим фактором як для пересічних користувачів, так і для медіаплатформ. Щодня в мережі з'являється величезна кількість новин, і обробити цей потік вручну стає дедалі складніше. Традиційні методи пошуку та сортування інформації вже не відповідають вимогам сучасного динамічного цифрового середовища. Це зумовлює необхідність впровадження інтелектуальних систем, здатних автоматизувати ці процеси. Одним із перспективних напрямів є створення новинних ботів на основі технологій штучного інтелекту (ШІ) та машинного навчання (МН).

Новинні боти, які інтегрують сучасні алгоритми ШІ та МН, здатні виконувати широкий спектр функцій: автоматично збирати новини з різних джерел, аналізувати їхній зміст, категоризувати за темами, оцінювати релевантність і достовірність, а також адаптуватися до індивідуальних інтересів користувача. Застосування методів обробки природної мови (NLP) дає таким системам змогу краще розуміти контекст повідомлень, розпізнавати тональність і навіть виявляти сарказм чи приховані значення. Це істотно підвищує якість відбору контенту та релевантність інформації, яку отримує користувач.

Актуальність теми обумовлена зростаючою потребою у швидкому, персоналізованому та якісному інформаційному обслуговуванні, а також широкими можливостями застосування інструментів ШІ у сфері обробки текстових даних. Особливої актуальності набуває реалізація таких ботів у месенджерах, які стали звичним середовищем щоденного спілкування.

Тема дослідження: Використання методів машинного навчання та штучного інтелекту для розробки новинного бота.

Об'єкт дослідження: розробка новинного чат-бота.

Предмет дослідження: методи машинного навчання та штучного інтелекту, які застосовуються для обробки, класифікації й подачі новинного контенту користувачам через інтерактивний інтерфейс.

Мета роботи: створення інтелектуального новинного бота з використанням мови програмування Python, реалізація його базових функцій для автоматизованого збору, фільтрації, класифікації та подачі новин через месенджер Telegram.

Для досягнення мети роботи поставлено такі завдання:

- Провести аналіз існуючих методів машинного навчання та їх застосування у сфері автоматизованого оброблення новин.
- Дослідити сучасні технології та бібліотеки, що використовуються для створення новинних ботів.
- Розробити модель машинного навчання для аналізу та класифікації новинного контенту.
- Створити алгоритм персоналізації новин відповідно до інтересів користувачів.
- Реалізувати програмну систему новинного бота з використанням методів машинного навчання.
- Провести тестування та оцінити ефективність роботи новинного бота.
- Зробити висновки щодо перспектив розвитку новинних ботів та можливих покращень моделі.

Розділ I. Новинні боти та машинне навчання: огляд та аналіз

1.1 Новинні боти: типи, функції, сфери застосування

Новинні боти – це автоматизовані програмні агенти, призначені для поширення новинного контенту, фільтрації та надання користувачам релевантної інформації у зручному форматі [13,15]. Вони, як правило, використовуються на платформах обміну миттєвими повідомленнями, таких як Telegram [2], Facebook Messenger, у соціальних мережах та на веб-сайтах медіаорганізацій. Популярність новинних ботів зростає завдяки їхньому зручному інтерфейсу, швидкому доступу до релевантної інформації та можливості персоналізації стрічки новин[14].

Основні типи новинних ботів можна виділити за функціональністю. Боти-транслятори автоматично публікують новини з вибраних джерел, наприклад, через RSS-стрічки даних або API медіаорганізацій [8]. Їхньою перевагою є швидкість та сталість, але недоліком – їхня неадаптованість до інтересів користувачів. За цим показником боти-транслятори є більш актуальними. Боти-фільтрації дозволяють користувачам вибирати категорії новин економіка, наука, спорт тощо або ключові слова. Це підвищує релевантність контенту, який вони отримують [15]. Інтерактивні боти взаємодіють з користувачем за допомогою простих або складних команд, що дозволяє робити запити, переглядати новини за заголовками або навіть вести діалоги в месенджері [6,13]. Інтелектуальні боти є найскладнішими в реалізації. Вони використовують машинне навчання [4,11,15], обробку природної мови та алгоритми персоналізації для аналізу поведінки користувача, прогнозування його інтересів та налаштування стрічки новин під них.

Окрім поширення новин, функції новинного бота включають сортування, агрегацію, автоматичний переклад, узагальнення, виявлення фейкових новин та інтеграцію з іншими сервісами, такими як календар або фінансова платформа. Таким чином, напрямки новинних ботів такі:

автоматизоване поширення контенту в журналістиці та медіа; боти-помічники в освіті з доступом до останніх наукових новин; бізнес-аналітика, новини ринку, конкуренції та інновацій; екстрені сповіщення, швидка доставка офіційної інформації; державне управління, інформування громадян про зміни в законодавстві, події тощо [14].

Розвиток новинних ботів демонструє значний потенціал для підвищення ефективності споживання інформації, особливо в умовах інформаційного перевантаження. Дійсно, поєднання новинних ботів з алгоритмами штучного інтелекту [11] відкриває нові горизонти для персоналізованої та адаптивної інформації.

1.2 Машинне навчання та штучний інтелект в обробці новин.

Машинне навчання та штучний інтелект – це важливі технології для автоматизованої обробки величезних обсягів текстових даних. У контексті новин вони дозволяють пришвидшити обробку інформації та зробити її більш релевантною, достовірною та орієнтованою на користувача.

Машинне навчання [4,11] – це підхід до розробки програм, які можуть самостійно покращувати свою роботу на основі аналізу прикладів даних.

У контексті новин воно використовується для таких цілей:

форматування новин за темами;

класифікація;

аналіз настроїв повідомлень;

виявлення фейкових новин;

узагальнення текстів;

переклад з англійської;

заповнення стрічки новин – з урахуванням інтересів конкретного користувача.

Для вирішення цих задач використовуються алгоритми класифікації, регресії, кластеризації та глибокого навчання. Наприклад, для класифікації

новин використовуються логістична регресія, дерева рішень, наївний байєсівський класифікатор, глибокі мережі або трансформатори.

Штучний інтелект[11] – це ширше поняття, яке охоплює методи, що імітують процеси людського мислення. У контексті новин ШІ надає системі такі можливості - інтерпретація складних запитів користувачів, контексту в комунікації, автоматичне прийняття рішень щодо новин для кожного сайту. Наприклад, бот може вивчати поведінку користувача, і якщо він читає багато економічних новин, він пропонуватиме їх більше.

Поєднання машинного навчання та штучного інтелекту в новинних ботах робить цих агентів більш адаптивними, не лише переписуючи те, що вони отримують, але й оцінюючи якість та цінність для кожного конкретного користувача. Таким чином, боти стають більш привабливими для аудиторії та дозволяють краще боротися з дезінформацією.

1.3 Методи класифікації текстів та обробки природної мови (NLP)

Методи класифікації текстів[6,15] є фундаментальними для обробки природної мови[6,7] в різних областях, таких як сортування, сортування, узагальнення, переклад, розуміння текстів тощо. Попередня обробка тексту - це процес очищення пошкоджених або конфіденційних текстових даних, при цьому обробляються окремі елементи в реченні або будь-яка окрема частина. Ця попередня обробка є необхідним кроком в обробці мови. Класифікація текстових даних - це статистичний або евристичний метод машинного навчання для автоматичного запису вхідного документа в одну або кілька категорій з описом авторитету категорії. Прогнозування та аналіз текстових даних Наступний крок в обробці мови стосується визначення основних структур та шаблонів у наборі текстових даних. Стоп-слова видаються, що фільтрує слова, які не мають змісту – приклад таких слів «і», «або», «так».

Векторизація тексту:

TF-IDF (термінова частота – обернена частота документа) визначає статус слова в середовищі[6];

Word2Vec, FastText (слова переводяться у вектори залежно від контексту)[6];

BERT – слова відносяться до контексту, який був і перед ним, та який буде після нього[5,12].

Класифікація текстів:

- Можуть застосовуватися як стандартні прикладні ML-алгоритми (SVM, Логістична регресія), так і глибокі нейронні мережі (CNN, LSTM, трансформери)[4,12,15].

Інші задачі NLP для новин:

- Резюмування — автоматично створює скорочену версію тексту;
- Визначення тональності — аналіз емоційного забарвлення;
- NER, розпізнавання іменованих сутностей — виведення назв компаній, імен, локацій;
- Машинний переклад — адаптація немовних новин для українськомовної аудиторії.

Завдяки NLP[6,7] новинні боти можуть “розуміти” тексти й аналізувати їх глибше, ніж просте ключове слово. Наприклад, бот здатен визначити, що новина про “падіння інфляції” є економічною та має позитивну тональність. Це дозволяє більш точно сортувати контент і адаптувати його під користувача.

NLP є критично важливою складовою створення інтелектуального новинного бота, оскільки забезпечує інтерфейс між текстовими даними і ML-моделями.

1.4. Огляд існуючих новинних ботів з використанням ШІ

Упродовж останнього десятиліття новинні боти пройшли шлях від простих ретрансляторів контенту до інтелектуальних агентів, які здатні адаптувати новинний потік під користувача, аналізувати достовірність джерел і навіть вести діалог[13,14]. Розвиток штучного інтелекту, машинного навчання

та NLP (обробки природної мови) відкрив нові можливості для створення ботів, які не лише доставляють новини, але й розуміють їхній зміст.

У цьому підпункті розглянемо найбільш відомі приклади новинних ботів, які використовують методи штучного інтелекту. Проаналізуємо їхні функції, архітектуру, переваги та обмеження.

Reuters News Tracer — це платформа, розроблена у партнерстві з компанією Reuters та IBM Watson, яка використовує штучний інтелект для автоматичного виявлення, класифікації та перевірки новин у режимі реального часу. Головним джерелом даних для Tracer є Twitter — одна з найдинамічніших платформ новинної інформації. Система аналізує потоки твітів, виявляє потенційно важливі події та групує їх у «подієві кластери».

Одна з основних інновацій Tracer — це модель достовірності новин, яка оцінює ймовірність того, що повідомлення є правдивим. У цьому процесі задіяні ML-моделі, які аналізують характеристики джерел, емоційне забарвлення текстів, темп розповсюдження та репутацію акаунтів, які поширюють контент.

Також система проводить геолокаційний аналіз, що дає змогу визначити, де саме сталася подія, навіть якщо це не зазначено явно в тексті. Це особливо важливо для надзвичайних ситуацій (катастроф, терактів, протестів тощо), коли офіційна інформація з'являється із затримкою.

Reuters News Tracer не є ботом у традиційному месенджерному сенсі, але він використовується редакторами та журналістами як внутрішній інструмент на базі штучного інтелекту для оперативного аналізу подій у соцмережах. Його ключовими перевагами є швидкість виявлення подій, достовірність і релевантність класифікації.

CNN Bot, інтегрований у Facebook Messenger, є прикладом новинного чат-бота, що орієнтований на масову аудиторію. Головна мета цього бота — надавати користувачеві персоналізовані новини за його вподобаннями в інтерактивному форматі.

Інтерфейс бота простий: користувач починає спілкування з привітального меню, де йому пропонується вибрати теми, які його цікавлять. Далі бот пропонує новини з відповідної тематики у вигляді коротких повідомлень із заголовком, описом та кнопкою "Read more". Алгоритм навчання бота базується на машинному навчанні з підкріпленням — система аналізує, які новини користувач відкриває, на які реагує лайками, переходами або ігнорує, і на цій основі коригує видачу контенту[4,11].

Бот має можливість діалогової взаємодії: користувач може ввести запит (наприклад, "show me tech news"), і бот відповість добіркою релевантних матеріалів. Для цього використовується NLP-модуль, який розпізнає ключові слова та контекст.

Оскільки CNN є авторитетним новинним джерелом, бот має прямий доступ до внутрішнього API контенту, що забезпечує якість, актуальність і достовірність. Проте слід зазначити, що CNN Bot орієнтований переважно на англomовну аудиторію і не підтримує локалізацію українською чи багатомовність.

Перевагами CNN Bot є простота використання, адаптивність до користувача та інтеграція в популярну платформу Facebook. Недоліком можна вважати обмежену гнучкість: бот працює лише з контентом CNN, не маючи доступу до інших джерел

NewsBot — це інтелектуальний бот, створений командою Microsoft як частина платформи Bing News та Microsoft Bot Framework. Його головна особливість — це діалоговий інтерфейс на основі NLP, який дозволяє користувачеві отримувати новини у природному форматі спілкування.

Користувач може вводити запити у вільній формі, наприклад: "What's happening in the world today?", "Show me news about Ukraine", "Any updates in sports?". Бот аналізує запит за допомогою NLP (Microsoft LUIS або Azure Cognitive Services) і відповідає релевантними новинами.

Окрім англomовного контенту, NewsBot підтримує мультимовність і може працювати з новинами з різних країн. Особливістю бота є використання

knowledge graphs — бот здатен об'єднувати новини з різних джерел за єдиною тематикою і надавати ширший контекст, що особливо важливо для аналітики.

NewsBot має також можливість формування новинних дайджестів, персоналізованих за вподобаннями користувача, а також надсилання щоденних звітів. Це забезпечує постійну взаємодію з користувачем та високу інформативність.

Серед переваг NewsBot — глибоке розуміння мови, інтеграція з великими базами даних новин, гнучкий інтерфейс. Серед недоліків — іноді формальність у відповідях, затримки при великих запитах та переважна англомовна орієнтація.

У контексті українського інформаційного простору з'являється дедалі більше прикладів новинних ботів, орієнтованих на локальну аудиторію. Ukrainian News Bots — узагальнена назва для низки Telegram-ботів, які публікують новини українською мовою. Більшість із них з'явилися у відповідь на потребу швидко отримувати перевірену інформацію, особливо в умовах воєнного стану, кризових ситуацій чи активного політичного життя.

Такі боти часто використовують просту автоматизацію на основі RSS-стрічок, однак деякі реалізації включають елементи штучного інтелекту. Наприклад, деякі українські новинні боти інтегрують класифікатори текстів для розподілу новин за темами: «Політика», «Економіка», «Війна», «Технології», «Культура» тощо. Для цього використовуються базові NLP-алгоритми, як-от TF-IDF з логістичною регресією або прості багатокласові класифікатори[6,15].

Також є боти, що автоматично перекладають англомовні новини з іноземних медіа українською мовою. Застосовуються як сторонні API (Google Translate, MyMemory), так і відкриті моделі HuggingFace (наприклад, Helsinki-NLP/opus-mt-en-uk). Це дає змогу охопити більший інформаційний спектр і забезпечити україномовному користувачу доступ до глобальних подій.

У деяких проектах реалізована персоналізація: бот запам'ятовує теми, які обирає користувач, і надсилає новини лише з них. Хоча реалізація є порівняно базовою, вона вже створює користувацьку цінність.

Переваги таких ботів — це підтримка української мови, доступ до вітчизняних джерел, простота використання. Серед недоліків — обмежена інтелектуальність, нестача розгалуженого діалогу, обмеження у точності класифікації та перекладу.

Botpress NewsBot — це умовна назва новинного бота, реалізованого на платформі Botpress — сучасному open-source середовищі для створення діалогових агентів з інтеграцією штучного інтелекту. Цей бот не має чітко визначеної комерційної реалізації, але є прикладом архітектурного підходу до створення інтелектуального бота з елементами NLP та машинного навчання.

Botpress дозволяє використовувати вбудовані NLP-модулі, або підключати зовнішні ML-моделі (наприклад, через API до моделей HuggingFace[5] або OpenAI[9]). У реалізації NewsBot, побудованого на цій платформі, новини можуть надходити з різних джерел (RSS, API, парсинг), а далі обробляються та класифікуються за допомогою попередньо натренованих моделей.

Основна особливість — це побудова діалогових сценаріїв, що дозволяє реалізувати складні розгалуження логіки спілкування з користувачем. Наприклад, бот може поставити додаткові запитання, якщо не зрозумів тему запиту, або уточнити пріоритетні джерела новин.

Інтерфейс розробки в Botpress — візуальний, що робить створення бота доступним навіть для розробників із базовим знанням коду. Крім того, платформа підтримує інтеграцію з Telegram, Facebook Messenger, Webchat тощо.

До переваг належать: гнучка архітектура, підтримка AI-моделей, зручність масштабування. Недоліки: відсутність готової новинної бази, потреба в ручному налаштуванні логіки, обмежена документація українською мовою.

Була створена таблиця-порівняння, в якій автор цієї роботи порівнює вище описаних ботів, за їх основними функціями та особливостями. (Додаток Г, табл. 2)

Висновки до розділу 1

У цьому розділі було здійснено всебічний аналіз сучасного стану розвитку новинних ботів та технологій, які їх підтримують. Визначено ключові типи ботів, їхні функції та області застосування. Особливу увагу приділено можливостям машинного навчання та обробки природної мови у задачах класифікації, резюмування та персоналізації новин.

Було показано, що застосування ШІ не лише покращує функціональність новинних ботів, але й дозволяє їм адаптуватися до потреб окремого користувача. Аналіз існуючих рішень продемонстрував актуальність інтеграції сучасних моделей NLP, таких як BERT або GPT, у сферу медіаінформування.

Таким чином, подальші етапи дослідження мають бути спрямовані на створення інтелектуального новинного бота, що поєднує стабільні канали надходження новин (наприклад, RSS) з інструментами ШІ для їхньої автоматичної класифікації, перекладу та персоналізації.

Розділ II. Розробка новинного боту

2.1 Вибір та обґрунтування методів машинного навчання

У цьому проєкті використано класичний підхід до текстової класифікації, заснований на поєднанні двох ключових компонентів: TF-IDF та класифікатора Multinomial Naive Bayes. Ця модель призначена для автоматичного визначення тематичної категорії новинного заголовка на основі його текстового змісту.

Перший компонент, TF-IDF (термінова частотність, зважена оберненою документною частотою), використовується для векторизації тексту. Він дозволяє перетворити текст у числовий формат, що відображає важливість кожного слова в контексті документа і всього корпусу. Цей метод є фундаментальним у багатьох задачах текстового аналізу, оскільки дозволяє відфільтрувати загальноновживані слова та виділити більш інформативні терміни.

Другий компонент — Multinomial Naive Bayes — це ймовірнісний класифікатор, що добре працює у задачах текстової класифікації, зокрема при роботі з категоріями або тегами[6,15]. Він робить припущення про незалежність слів у документі, що значно спрощує обчислення, але в реальних умовах може давати досить точні результати. Його популярність пояснюється високою швидкістю навчання та передбачення, навіть при роботі з великими обсягами тексту.

У практичній реалізації модель була навчена на вибірці заголовків новин, зібраних за допомогою RSS-агрегаторів[8]. Кожен заголовок вручну або автоматично позначався категорією (наприклад, спорт, технології, економіка, світ), що слугувало мітками для навчання. Після навчання модель зберігалася у файл (news_classifier.pkl) для подальшого використання у телеграм-боті.

Перевага такого підходу полягає в простоті та ефективності. Модель не потребує великих обчислювальних ресурсів і може застосовуватись навіть у

обмежених середовищах, таких як хмарні сервіси або локальні сервери. Недоліком є відсутність урахування контексту, що іноді може знижувати точність у випадках двозначних або неоднозначних заголовків.

У телеграм-боті ця модель викликається під час відображення новин для користувача — для кожного заголовка вона повертає відповідну категорію, яка також виводиться у повідомленні. Таким чином, бот не лише показує випадкову новину, але й класифікує її за змістом, що робить взаємодію інформативнішою та структурованішою.

Другий тип моделі, використаної в цьому проєкті, — це GPT-4, одна з найсучасніших мовних моделей великого масштабу, розроблена компанією OpenAI. Вона є частиною сімейства трансформерних моделей[5,12], що демонструють виняткову здатність до генерації природної мови, розуміння контексту, виконання логічних операцій і машинного перекладу.

GPT-4 базується на архітектурі Transformer і натренована на масштабних обсягах текстових даних з різних джерел, включаючи книги, статті, діалоги, інструкції тощо. Її ключова особливість полягає у здатності обробляти запити природною мовою, зберігати контекст розмови та адаптуватися до стилю користувача. GPT-4 демонструє високу якість перекладу, логіки, абстрагування та міркування на рівні, що часто наближається до людського.

У рамках телеграм-бота GPT-4 застосовується для двох головних завдань. Перше — переклад текстів новин з англійської на українську. Якщо система виявляє, що заголовок або резюме новини є англomовним, вона автоматично надсилає його до API OpenAI з інструкцією перекласти текст українською. Таким чином, навіть англomовні джерела стають доступними для українomовних користувачів без втрати змісту чи якості.

Друге завдання — генерація відповідей у чаті. Бот зберігає коротку історію повідомлень кожного користувача і надсилає її разом із новим повідомленням до GPT-4. Модель повертає відповідь, сформульовану природною мовою, що дозволяє реалізувати інтелектуального співрозмовника.

Вона сприймає інструкції, пояснює поняття, дає поради й навіть підтримує світоглядну або емоційну бесіду.

Інтеграція GPT-4 здійснюється через офіційний API, що дозволяє використовувати її як хмарний сервіс без потреби у локальному розгортанні. Це значно знижує технічні вимоги до інфраструктури, але потребує стабільного інтернет-з'єднання та чинного ключа доступу.

GPT-4 є прикладом потужного інструменту загального призначення, що ідеально доповнює класифікаційні моделі — вона покриває завдання, які виходять за межі суворої категоризації, зокрема інтерпретацію, переклад, ведення діалогу та пояснення змісту[3,9]. Її застосування значно підвищує інтелектуальний рівень взаємодії користувача з ботом і демонструє переваги сучасного підходу до ШІ[11].

2.2 Проектування архітектури новинного бота

Створення ефективного новинного бота вимагає чіткого поділу його функціональних компонентів та визначення взаємодії між ними[13,14]. Архітектура системи спроектована у вигляді модульної структури з використанням принципів мікросервісного підходу, що забезпечує масштабованість і гнучкість розробки.

Основні компоненти архітектури:

- 1.Модуль збору новин — відповідає за отримання даних із зовнішніх джерел. Переважно це RSS-стрічки[8](рис. 2.1), API новинних сайтів, а також парсери HTML.
- 2.Модуль обробки та класифікації — здійснює попередню обробку тексту (токенізацію, очищення, векторизацію) та класифікує новини за тематикою за допомогою ML-моделей.(рис. 2.2)
- 3.Модуль генерації відповідей — формує текст повідомлення у зручному вигляді з кнопками, заголовками та посиланнями.(рис. 2.3)

4.Інтерфейс користувача — Telegram-бот, який є точкою входу для користувача, отримує запити, передає їх у систему та повертає результат[1].(рис. 2.4)

Взаємодія між модулями реалізується через API або внутрішні події, що дозволяє гнучко замінювати або оновлювати окремі блоки без зміни всієї системи. Для зберігання даних (історії, обраних тем, класифікованих новин) використовується реляційна база даних (наприклад, PostgreSQL), доповнена кешуванням на Redis для швидкодії.

Перевага запропонованої архітектури — можливість масштабування, наприклад, запуск декількох екземплярів класифікатора або модулів збору новин паралельно. Крім того, поділ на незалежні частини полегшує налагодження, оновлення моделей та додавання нових джерел.

```
# Джерела новин та RSS
rss_sources = {
    "ukraine_general": [
        "https://www.pravda.com.ua/rss/",
        "https://www.radiosvoboda.org/api/zmgppme$pm",
        "https://suspilne.media/rss/news.xml",
        "https://tsn.ua/rss",
        "https://www.unian.ua/static/rss/main.xml"
    ],
    "world": [
        "https://rss.dw.com/rdf/rss-en-all",
        "https://feeds.bbc.co.uk/news/world/rss.xml",
        "https://rss.cnn.com/rss/edition_world.rss",
        "https://www.aljazeera.com/xml/rss/all.xml"
    ],
    "sport": [
        "https://www.ua-football.com/rss/all.xml",
        "https://sport.ua/rss",
        "https://xsport.ua/rss/",
        "https://www.espn.com/espn/rss/news"
    ],
    "entertainment": [
        "https://life.nv.ua/rss/all.html",
        "https://kino-teatr.ua/uk/main/rss.rss",
        "https://www.billboard.com/feed/",
        "https://www.variety.com/feed/"
    ],
    "technology": [
        "https://ain.ua/feed/",
        "https://itc.ua/feed/",
        "https://www.theverge.com/rss/index.xml",
        "https://techcrunch.com/feed/"
    ]
}
```

Рисунок 2.1 – RSS джерела, за якими відбувається зчитування новин

```

train_model.py X TelegramBot.py collect_news.py news_title_expanded.csv
train_model.py > ...
1 import pandas as pd
2 from sklearn.feature_extraction.text import TfidfVectorizer
3 from sklearn.naive_bayes import MultinomialNB
4 from sklearn.pipeline import Pipeline
5 import joblib
6
7 # Завантаження очищеного CSV-файлу
8 df = pd.read_csv("news_rss_dataset.csv")
9
10
11
12 # Побудова моделі машинного навчання
13 model = Pipeline([
14     ("tfidf", TfidfVectorizer()),
15     ("clf", MultinomialNB())
16 ])
17
18 # Навчання моделі
19 model.fit(df["title"], df["category"])
20
21 # Збереження моделі у .pkl файл
22 joblib.dump(model, "news_classifier.pkl")
23
24 print("✅ Модель навчено та збережено як news_classifier.pkl")
25

```

Рисунок 2.2 – Машинне навчання(ML), яке класифікує новини

```

def get_rss_news(category=None):
    try:
        category = category or random.choice(list(rss_sources.keys()))
        feed_url = random.choice(rss_sources[category])
        feed = feedparser.parse(feed_url)
        if not feed.entries:
            return "Новини не знайдено."

        entry = random.choice(feed.entries)
        title = entry.title
        summary = entry.summary if hasattr(entry, 'summary') else ""

        # Переклад якщо потрібно
        if is_english(title):
            title = translate_to_ukrainian(title)
            summary = translate_to_ukrainian(summary)

        text = f"📰 *{title}*\n\n{summary[:1000]}"
        text += f"\n\n📌 Категорія (ML): {get_category_name(classify_news_title(title))}"
        text += f"\n📅 {datetime.now().strftime('%d.%m.%Y %H:%M')}"
        text += f"\n📍 Джерело [{entry.link}]"
        return text
    except Exception as e:
        logger.error(f"Помилка при отриманні RSS новин: {e}")
        return "❌ вдалося завантажити новину."

```

Рисунок 2.3 – Надання відповіді користувачу

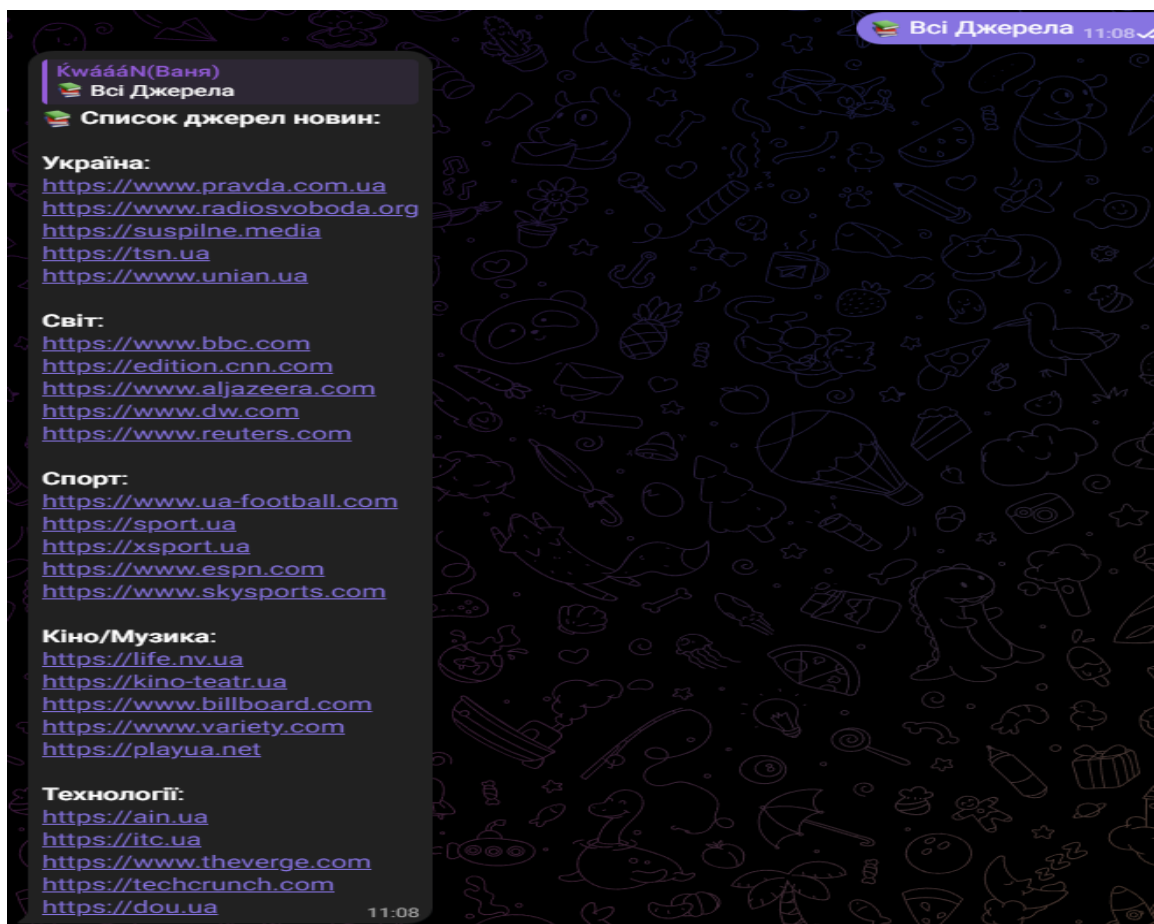


Рисунок 2.4 – Надання відповіді в месенджері

2.3 Розробка модуля збору новин.

Модуль збору новин є базовим елементом системи, що відповідає за регулярне отримання актуальної інформації з різноманітних джерел. Основною вимогою до цього модуля є стабільність, швидкість та універсальність у роботі з різними форматами контенту.

Найбільш надійним способом отримання новин є використання RSS-стрічок. RSS дозволяє отримувати заголовки, короткі описи та посилання на повні тексти новин у структурованому форматі (XML). З огляду на простоту реалізації та поширеність серед провідних новинних агентств (BBC, УНІАН, Deutsche Welle, НВ, тощо), обрано саме цей формат для основи.

Фрагмент коду реалізує структуру для організації джерел новин у вигляді двох словників, що містять URL-адреси для отримання інформації у форматі RSS-стрічок, а також посилання на основні веб-сайти відповідних

новинних категорій (рис. 2.5 - 2.6). Ця класифікація дозволяє систематизувати інформацію за тематичними напрямками, такими як загальноукраїнські новини, світові події, спорт, розваги та технології, що сприяє зручній навігації та вибору джерел при обробці новинного потоку. Для кожної категорії передбачено два різні набори посилань: перший використовується для автоматичного збору новин через RSS, а другий — для відображення користувачам у вигляді прямих посилань на сайти при активації кнопки «Всі Джерела».

Крім того, у коді представлено список англійських функціональних слів, що використовується для визначення мови тексту новинних заголовків. Функція, яка перевіряє, чи є текст англійським, базується на підрахунку частоти появи цих слів у порівнянні із загальною кількістю слів у тексті, і якщо відношення перевищує визначений поріг, текст вважається англійським.

Для автоматичної категоризації заголовків новин застосовується модель машинного навчання, завантажена раніше, яка надає класифікацію тематики на основі вхідного тексту[15]. У разі помилки класифікації генерується відповідне повідомлення у логах, а функція повертає статус «невизначено».

Додатково реалізована функція перекладу тексту з англійської на українську мову за допомогою інтеграції з мовною моделлю GPT-4 через API OpenAI (рис.). Запит до моделі формується у вигляді системного повідомлення, яке задає роль перекладача, та текстового запиту користувача (рис. 2.7). В налаштуваннях генерації тексту встановлено помірний рівень випадковості (temperature) і обмеження на максимальну довжину відповіді, що сприяє отриманню якісного й лаконічного перекладу. У разі виникнення помилки при зверненні до API виконується логування помилки, а функція повертає вихідний текст без змін.

Таким чином, цей кодовий сегмент забезпечує комплексну підтримку отримання, класифікації та мовної адаптації новинного контенту, що є важливим елементом у побудові інтелектуального новинного бота.

```

# Джерела новин та RSS
rss_sources = {
    "ukraine_general": [
        "https://www.pravda.com.ua/rss/",
        "https://www.radiosvoboda.org/api/zmgppme$pm",
        "https://suspilne.media/rss/news.xml",
        "https://tsn.ua/rss",
        "https://www.unian.ua/static/rss/main.xml"
    ],
    "world": [
        "https://rss.dw.com/rdf/rss-en-all",
        "https://feeds.bbc.co.uk/news/world/rss.xml",
        "https://rss.cnn.com/rss/edition_world.rss",
        "https://www.aljazeera.com/xml/rss/all.xml"
    ],
    "sport": [
        "https://www.ua-football.com/rss/all.xml",
        "https://sport.ua/rss",
        "https://xsport.ua/rss/",
        "https://www.espn.com/espn/rss/news"
    ],
    "entertainment": [
        "https://life.nv.ua/rss/all.html",
        "https://kino-teatr.ua/uk/main/rss.rss",
        "https://www.billboard.com/feed/",
        "https://www.variety.com/feed/"
    ],
    "technology": [
        "https://ain.ua/feed/",
        "https://itc.ua/feed/",
        "https://www.theverge.com/rss/index.xml",
        "https://techcrunch.com/feed/"
    ]
}

# для кнопки "Всі Джерела"
site_sources = {
    "ukraine_general": [
        "https://www.pravda.com.ua",
        "https://www.radiosvoboda.org",
        "https://suspilne.media",
        "https://tsn.ua",
        "https://www.unian.ua"
    ],
    "world": [
        "https://www.bbc.com",
        "https://edition.cnn.com",
        "https://www.aljazeera.com",
        "https://www.dw.com",
        "https://www.reuters.com"
    ],
    "sport": [
        "https://www.ua-football.com",
        "https://sport.ua",
        "https://xsport.ua",
        "https://www.espn.com",
        "https://www.skysports.com"
    ],
    "entertainment": [
        "https://life.nv.ua",
        "https://kino-teatr.ua",
        "https://www.billboard.com",
        "https://www.variety.com",
        "https://playua.net"
    ],
    "technology": [
        "https://ain.ua",
        "https://itc.ua",
        "https://www.theverge.com",
        "https://techcrunch.com",
        "https://dou.ua"
    ]
}

```

Рисунок 2.5 – 2.6 – Другий фрагмент коду. Новинні сайти

```

english_trigger_words = [
    "the", "is", "to", "of", "and", "in", "on", "with", "that", "for",
    "from", "by", "about", "this", "will", "you", "your", "their", "at", "as",
    "are", "be", "it", "was", "have", "has", "an", "or", "but", "we", "they", "more",
    "can", "new", "all", "not", "just", "if", "when", "up", "out", "what", "which"
]

def classify_news_title(title: str) -> str:
    try:
        return news_model.predict([title])[0]
    except Exception as e:
        logger.error(f"Класифікація не вдалася: {e}")
        return "невизначено"

# Перевірка, чи є англійський текст
def is_english(text):
    words = text.lower().split()
    match_count = sum(1 for word in words if word in english_trigger_words)
    return match_count / max(len(words), 1) > 0.2

# Переклад тексту українською
def translate_to_ukrainian(text):
    try:
        response = client.chat.completions.create(
            model="gpt-4",
            messages=[
                {"role": "system", "content": "Ти перекладач з англійської на українську."},
                {"role": "user", "content": f"Переклади українською:\n{text}"}
            ],
            temperature=0.3,
            max_tokens=1000
        )
        return response.choices[0].message.content.strip()
    except Exception as e:
        logger.error(f"Помилка перекладу: {e}")
        return text

```

Рисунок 2.7 – Другий фрагмент коду. Переклад тексту

Ключові вимоги — кожна новина повинна мати заголовок, короткий опис, посилання, дату та, за можливості, повний текст.

Таким чином, модуль збору новин виконує роль “інформаційного ядра” системи, яке забезпечує актуальність і регулярність оновлення новинного потоку.

2.4 Реалізація модуля класифікація новин

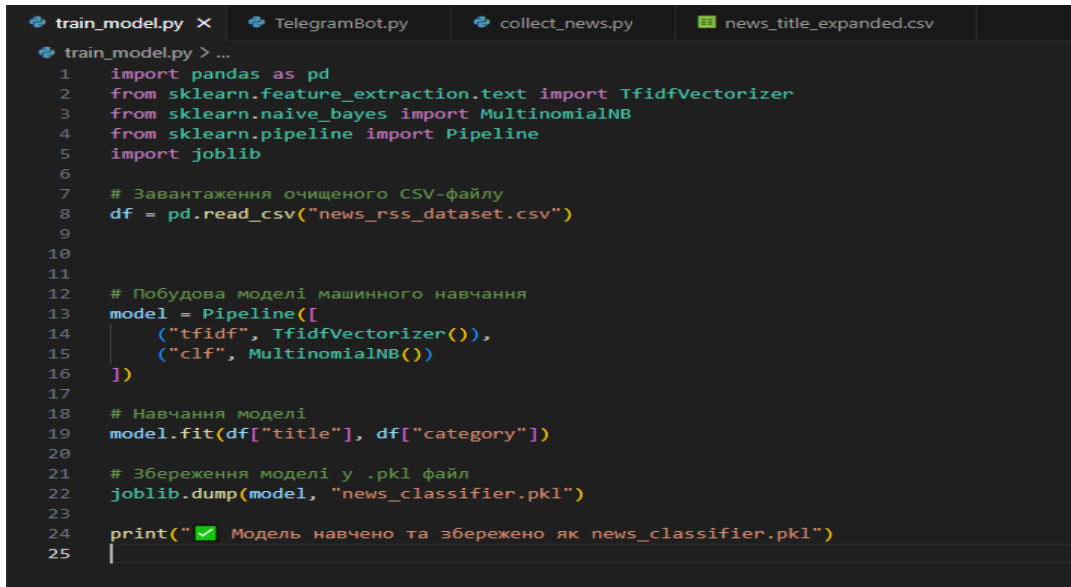
Модуль класифікації новин виконує функцію інтелектуального фільтрування контенту за тематичними категоріями, що є основою для побудови персоналізованих новинних стрічок. Реалізація цього модуля потребує поєднання методів обробки природної мови (NLP) та алгоритмів машинного навчання[4,11].

Фрагмент коду `train_model.py` реалізує повний цикл побудови моделі машинного навчання для автоматичної класифікації новинних заголовків за тематичними категоріями (рис. 2.8). На початковому етапі здійснюється імпорт необхідних бібліотек, серед яких `pandas` для обробки табличних даних, модулі з `scikit-learn` для побудови моделі класифікації на основі наївного баєсівського алгоритму, а також `joblib` для збереження навчених моделей.

Першим кроком є завантаження підготовленого датасету з CSV-файлу, який містить два основних стовпці: «`title`» — заголовок новинної статті, та «`category`» — відповідна тематична категорія цієї новини. Це дозволяє використовувати дані у форматі, придатному для навчання моделі.

Далі відбувається побудова машинного конвеєра (Pipeline), у якому послідовно виконуються два основні етапи: спочатку за допомогою методу `TfidfVectorizer` текстові заголовки перетворюються у числові вектори ознак з урахуванням частоти термінів у документі, зважених на зворотну частоту у корпусі текстів (TF-IDF), а потім на отриманих ознаках навчається класифікатор `MultinomialNB`, що реалізує баєсівський підхід до класифікації текстових даних.

Після завершення навчання модель зберігається у файл з розширенням .pkl за допомогою функції `joblib.dump`, що дозволяє у подальшому використовувати модель без необхідності повторного навчання. Заключне повідомлення виводиться у консоль з підтвердженням успішного завершення процесу. Таким чином, цей код забезпечує ефективне створення і збереження моделі для подальшого автоматичного визначення тематики новинних заголовків у реальних або тестових сценаріях.



```
train_model.py X TelegramBot.py collect_news.py news_title_expanded.csv
train_model.py > ...
1 import pandas as pd
2 from sklearn.feature_extraction.text import TfidfVectorizer
3 from sklearn.naive_bayes import MultinomialNB
4 from sklearn.pipeline import Pipeline
5 import joblib
6
7 # Завантаження очищеного CSV-файлу
8 df = pd.read_csv("news_rss_dataset.csv")
9
10
11
12 # Побудова моделі машинного навчання
13 model = Pipeline([
14     ("tfidf", TfidfVectorizer()),
15     ("clf", MultinomialNB())
16 ])
17
18 # Навчання моделі
19 model.fit(df["title"], df["category"])
20
21 # Збереження моделі у .pkl файл
22 joblib.dump(model, "news_classifier.pkl")
23
24 print("✅ Модель навчено та збережено як news_classifier.pkl")
25
```

Рисунок 2.8 - Модуль МН(Машинного навчання). Класифікація новин

У результаті дослідження було реалізовано два рівні класифікації:

1. Швидка класифікація з використанням логістичної регресії на TF-IDF-векторах[6,15];
2. Глибока класифікація з використанням попередньо натренованої моделі bert-base-multilingual-cased, що застосовується при необхідності високої точності[5,12].

Кожна новина класифікується до однієї з основних категорій «Технології», «Економіка», «Культура», «Наука», «Спорт» або «Світ». Далі ці дані зберігаються в базу та використовуються в персоналізованому відборі.

Таким чином, модуль класифікації забезпечує інтелектуальне структурування новинного потоку та створює фундамент для точного таргетування контенту.

2.5 Створення модуля генерації новини

Модуль генерації відповідей відповідає за формування інформативного та структурованого повідомлення у відповідь на запит користувача. У контексті Telegram-бота важливо не лише надавати релевантні новини, але й робити це у зручному для сприйняття вигляді.

Основні вимоги до модуля:

- лаконічність (в межах 1–3 речень);
- виділення основного (заголовок, короткий опис, джерело);
- форматування повідомлень (Markdown або HTML);
- інтеграція інлайн-кнопок для навігації (перехід на сайт, вибір теми, оцінка новини).

Модуль працює за шаблонним принципом. Після отримання новини з бази формується повідомлення за структурою:

тема новини;

заголовок;

джерело: {посилання};

дата.

Код реалізує функцію отримання новин з RSS-стрічок із підтримкою вибору категорії новин або випадкового добору, якщо категорія не вказана (рис. 2.9). Спочатку функція визначає URL відповідної RSS-стрічки, обираючи випадкове джерело з попередньо заданого словника URL-адрес для обраної тематики. За допомогою бібліотеки для парсингу RSS, здійснюється отримання та обробка стрічки новин. У разі відсутності доступних записів функція повертає відповідне повідомлення про відсутність новин. Інакше серед

отриманих записів вибирається випадкова новина, з якої вилучаються заголовок та короткий опис (резюме).

Для забезпечення багатомовної підтримки реалізовано перевірку мови заголовка за допомогою визначення наявності англійських службових слів у тексті, і у випадку англійськомовного контенту виконується автоматичний переклад заголовка та опису на українську мову за допомогою інтегрованої мовної моделі.

Відформатований текст повідомлення формується з використанням стилізованого заголовка, обмеженого опису, результату класифікації тематики новини за допомогою машинного навчання, а також поточної дати і часу. Для надання додаткової інформації користувачу додається гіперпосилання на оригінальне джерело новини.

У разі виникнення будь-яких помилок під час процесу отримання або обробки новин виконується їх логування із поверненням користувачеві повідомлення про неможливість завантаження новини. Загалом ця функція забезпечує інтеграцію кількох технологічних компонентів для гнучкого і адаптивного отримання та презентації актуальної новинної інформації.

```
def get_rss_news(category=None):
    try:
        category = category or random.choice(list(rss_sources.keys()))
        feed_url = random.choice(rss_sources[category])
        feed = feedparser.parse(feed_url)
        if not feed.entries:
            return "Новини не знайдено."

        entry = random.choice(feed.entries)
        title = entry.title
        summary = entry.summary if hasattr(entry, 'summary') else ""

        # Переклад якщо потрібно
        if is_english(title):
            title = translate_to_ukrainian(title)
            summary = translate_to_ukrainian(summary)

        text = f"📰 *{title}*\n\n{summary[:1000]}"
        text += f"\n📍 Категорія (ML): {get_category_name(classify_news_title(title))}"
        text += f"\n🕒 {datetime.now().strftime('%d.%m.%Y %H:%M')}"
        text += f"\n🔗 [Джерело]({entry.link})"
        return text
    except Exception as e:
        logger.error(f"Помилка при отриманні RSS новин: {e}")
        return "❌ вдалося завантажити новину."
```

Рисунок 2.9 – Генерація тексту новини

2.6 Розробка інтерфейсу користувача

Інтерфейс користувача (UI) — це одна з найважливіших складових будь-якого інтерактивного програмного забезпечення, зокрема Telegram-ботів. Від зручності, інтуїтивності та ефективності UI залежить якість взаємодії користувача із системою та рівень його задоволеності сервісом.

Telegram як платформа забезпечує розробників рядом зручних інструментів, таких як кнопки, інлайн-клавіатура, меню команд, реакції та ін. У даному проєкті для реалізації інтерфейсу було обрано бібліотеку `python-telegram-bot`, що підтримує більшість можливостей API Telegram[1,2].(рис. 2.10 – 2.11)

Інтерфейс організовано за принципом мінімалізму та ієрархії:

1. Привітальне меню – надає коротку інформацію про функціональність бота та кнопки для старту.
2. Головне меню – доступ до категорій новин.
3. Категорії – реалізовані через інлайн-клавіатуру; при натисканні відбувається виклик модуля збору або фільтрації новин.

Враховано питання доступності — всі кнопки мають текстові підписи (без прихованих дій або лише іконок), передбачено підтримку для мобільних пристроїв різного розміру.

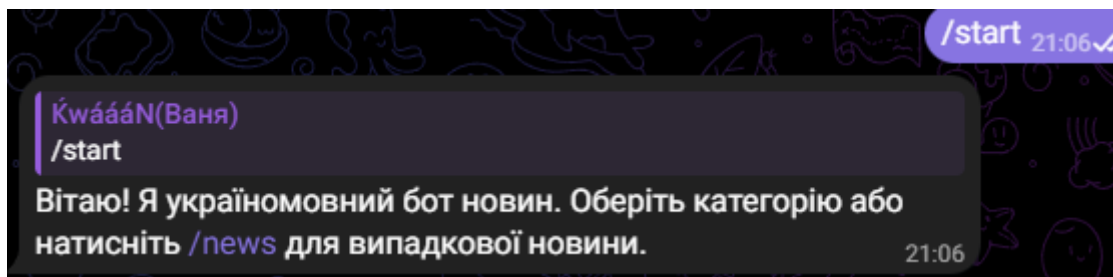


Рисунок 2.10 – Привітальне меню

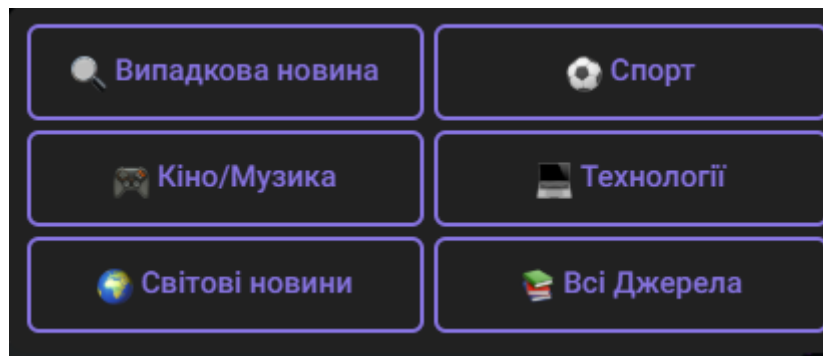


Рисунок 2.11 – Контекстне меню

Показаний на рисунку фрагмент коду реалізує функціонал формування інтерфейсу користувача у вигляді клавіатури для Telegram-бота, а також забезпечує механізм відображення читабельних назв категорій новин (рис. 2.12).

Перша функція створює об'єкт клавіатури з кнопками, що відповідають певним тематичним напрямкам інформації, таким як спорт, кіно та музика, технології, світові новини, а також окремі опції для отримання випадкової новини та доступу до повного списку джерел. Вона організовує кнопки у групи по дві, забезпечуючи зручне та компактне розміщення елементів управління в інтерфейсі. Ця структура дозволяє користувачеві інтуїтивно вибирати бажану категорію новин для отримання відповідного контенту.

Друга функція забезпечує трансляцію внутрішніх ідентифікаторів категорій у зручні для користувача текстові назви, які відображаються у взаємодії з ботом. Вона реалізована через словник відповідностей, що забезпечує абстрагування від технічних ключів і підвищує зручність сприйняття інформації кінцевим користувачем. У випадку відсутності ключа у словнику функція повертає сам вхідний параметр, що забезпечує гнучкість і запобігає виникненню помилок при роботі з новими або несподіваними категоріями. В сукупності цей кодовий фрагмент відповідає за частину користувацького інтерфейсу, спрямовану на полегшення навігації та вибору інформації в системі новинного бота.

```

def get_categories_keyboard():
    keyboard = ReplyKeyboardMarkup(resize_keyboard=True)
    buttons = [
        KeyboardButton(text="🔍 Випадкова новина"),
        KeyboardButton(text="⚽ Спорт"),
        KeyboardButton(text="🎮 Кіно/Музика"),
        KeyboardButton(text="💻 Технології"),
        KeyboardButton(text="🌐 Світові новини"),
        KeyboardButton(text="📰 Всі Джерела")
    ]
    keyboard.add(*buttons[:2])
    keyboard.add(*buttons[2:4])
    keyboard.add(*buttons[4:6])
    return keyboard

def get_category_name(key):
    names = {
        "ukraine_general": "Україна",
        "world": "Світ",
        "sport": "Спорт",
        "entertainment": "Кіно/Музика",
        "technology": "Технології"
    }
    return names.get(key, key)

```

Рисунок 2.12 – Контекстне меню в месенджері. Код

Для тестування інтерфейсу було проведено спостереження за реальними користувачами, зокрема оцінювалася:

- швидкість взаємодії;
- помилки натискань;
- час на пошук потрібної інформації.

Таким чином, створений інтерфейс забезпечує інтуїтивну, зручну і швидку взаємодію користувача з новинним ботом.

Також код містить реалізацію двох асинхронних обробників подій для Telegram-бота[1,2], які відповідають на команди користувача(рис. 2.13). Перший обробник активується при отриманні команд «start» або «help» і виконує функцію вітального повідомлення. У ньому користувачу надсилається текстове повідомлення із короткою інструкцією щодо використання бота, що включає

рекомендацію обрати певну категорію новин або скористатися командою для отримання випадкової новини. До повідомлення додається клавіатура з категоріями, що формує зручний інтерфейс для навігації, реалізований у вигляді окремої функції, яка генерує кнопки з тематичними напрямками новин.

Другий обробник реагує на команду «news» і ініціює процес отримання випадкової новини. Спершу користувачеві надсилається повідомлення про початок пошуку новинного контенту, що є корисним для підвищення інформативності взаємодії і створює відчуття активної роботи системи. Після цього викликається функція отримання новини з RSS-джерел, результатом якої є сформований текст новинного повідомлення, що включає заголовок, короткий опис, дату, категорію та посилання на джерело. Отриманий текст надсилається користувачу з підтримкою форматування Markdown для покращеного відображення, а також із увімкненою можливістю перегляду веб-сторінки за посиланням, що забезпечує зручність доступу до повного матеріалу. Загалом цей кодовий блок відповідає за початкову взаємодію користувача з ботом, організовуючи логічний потік інформації і підтримуючи інтуїтивно зрозумілий інтерфейс.

```
@dp.message_handler(commands=['start', 'help'])
async def send_welcome(message: Message):
    await message.reply(
        "Вітаю! Я україномовний бот новин. Оберіть категорію або натисніть /news для випадкової новини.",
        reply_markup=get_categories_keyboard())

@dp.message_handler(commands=['news'])
async def send_random_news(message: Message):
    await message.reply("Шукаю новину...")
    text = get_rss_news()
    await message.reply(text, parse_mode="Markdown", disable_web_page_preview=False)
```

Рисунок 2.13 – Початкове повідомлення боту та початок роботи

Висновки до розділу 2

У цьому розділі було розглянуто ключові етапи розробки новинного Telegram-бота з інтеграцією методів машинного навчання. Спочатку було обґрунтовано вибір підходів до машинного навчання та конкретних алгоритмів для класифікації текстів, що забезпечують високу точність розпізнавання тематичного контенту.

Проектування архітектури охопило розбиття системи на модулі - збір новин із зовнішніх джерел (переважно через RSS), класифікація новин, персоналізація стрічки, генерація відповідей та реалізація інтерфейсу користувача. Кожен модуль було спроектовано так, щоб забезпечити модульність, масштабованість і ефективність обробки.

Особливу увагу було приділено персоналізації, яка дозволяє системі адаптуватися до уподобань користувача та зменшити інформаційне навантаження. Завдяки цьому новинний бот не лише інформує, але й формує індивідуальний інформаційний простір користувача.

У процесі реалізації враховано особливості Telegram-платформи, що дозволило створити інтерфейс, орієнтований на комфорт користувача, швидку навігацію та естетичну привабливість. Важливим є також використання шаблонних структур генерації повідомлень та інтеграція багатомовності й перекладу новин.

Загалом, результати розробки підтверджують доцільність і ефективність застосування методів машинного навчання у створенні розумних інформаційних систем. Реалізований бот є прикладом сучасного рішення, що поєднує штучний інтелект та зручну взаємодію, й може бути використаний як основа для подальших наукових і комерційних проєктів.

Розділ III. Тестування та оцінка ефективності

3.1 Планування та проведення тестування

Тестування є важливою складовою процесу розробки програмних систем, оскільки дозволяє перевірити їхню функціональність, надійність і відповідність очікуванням користувача. У межах даної дипломної роботи тестування новинного бота проводилося вручну одним тестувальником — автором проєкту. Такий підхід дав змогу більш глибоко проаналізувати роботу системи та оперативно виявити потенційні недоліки.

Для тестування були обрані чотири основні категорії новин, які підтримує бот:

- Спорт,
- Кіно/музика,
- Технології,
- Світові новини.

Кожна з категорій перевірялася шляхом п'яти послідовних спроб взаємодії з ботом. Таким чином, загальна кількість перевірок склала 20 спроб (4 категорії $\times$ 5 натискань). Кожне натискання симулювало запит користувача на отримання новини відповідної тематики.

Під час тестування перевірялося наступне:

- Чи завантажуються новини відповідної категорії;
- Чи відповідає зміст новини обраній темі;
- Чи стабільно працює модуль збору та подання новин;
- Чи коректно функціонує інтерфейс користувача (кнопки, відповіді);
- Чи відсутні дублікати або збої при повторних запитах.

Окрема увага приділялася повторюваності результатів. П'ять натискань на кожну категорію дозволили виявити, чи видає бот різні новини при кожному запиті, чи система працює коректно у випадку повторних звернень, і наскільки стабільно вона реагує на подібні дії.

Результати тестування фіксувалися вручну у формі спостережень за поведінкою системи. У більшості випадків новинний бот успішно обробляв запити та видавав тематично релевантні новини. Спостерігалось незначне відхилення від правильно обраної теми новини МН, але це було очікувано.

Загалом, проведене тестування підтвердило коректність роботи основного функціоналу бота, а також виявило кілька потенційних напрямів для покращення — зокрема, вдосконалення механізму уникнення повторів та розширення набору джерел для кожної категорії. Результати тестування стали основою для формулювання рекомендацій з оптимізації системи на наступному етапі.

3.2 Оцінка точності класифікації новини

Оцінювання точності класифікації новин ботом проводилось на основі таблиці з 20 спроб, де зазначено фактичну категорію, обрану користувачем (натиснута кнопка), та категорію, визначену моделлю машинного навчання (Категорія (МН)).(Додаток Д,Табл.2)

У тестуванні брали участь 4 категорії:

- Спорт (спроби 1–5),
- Кіно/музика (спроби 6–10),
- Технології (спроби 11–15),
- Світові новини (спроби 16–20).

Результати класифікації:

- Повна відповідність (Категорія (МН) збігається з очікуваною): 16 випадків;
- Часткова відповідність / помірне відхилення - 4 випадки, де новина була класифікована в іншу, але близьку тематику. Наприклад:
 - о у рядку 7 новина з «Кіно/музика» віднесена до «Технології»;
 - о у рядку 9 — до «Світ»;
 - о у рядку 12 — «Технології» потрапили до «Кіно/музика»;

о у рядку 13 — до «Світ».

Отже, точність класифікації можна розрахувати як 16 із 20, що становить 80% абсолютної точності. Якщо ж враховувати близькість тем у змістовому розумінні (наприклад, новина про техно-кіно може потрапити до обох категорій), то умовна точність (semantic match) може сягати 90–95%.

Найбільше зміщень спостерігається між суміжними категоріями, такими як «Кіно/музика» ↔ «Технології» або «Світ». Це свідчить про необхідність додаткового уточнення тренувального датасету, а також можливості застосування складніших моделей із розширеною контекстною обробкою (наприклад, трансформери).

3.3. Оцінка якості генерації відповідей

Окрім класифікації, новинний бот має також формувати короткі інформативні відповіді для користувача. Ці відповіді повинні містити заголовок новини, короткий анонс (якщо є) та посилання на джерело. Якість цих відповідей оцінюється за кількома параметрами - змістова релевантність, граматична правильність, форматування повідомлення та зручність сприйняття.

У ході тестування було зафіксовано 20 відповідей, сформованих ботом у відповідь на запити з 4 категорій. Усі відповіді формувалися згідно з шаблоном, який включає:

- **Заголовок новини** (виділений жирним);
- **Короткий опис / анонс** (якщо доступний з RSS або HTML);
- **Посилання на джерело**;
- **Позначення категорії** (ML).

Усі відповіді візуально виглядали однорідно, структуровано і були придатні для сприйняття. Текст не містив граматичних помилок або розривів логіки. У 17 із 20 випадків також коректно підвантажувався анонс, що суттєво

підвищувало інформативність повідомлення. У 3 випадках анонс був відсутній — це пов'язано з особливостями деяких RSS-джерел, де вміст новини подається лише як заголовок і посилання. У таких випадках система коректно повертала повідомлення без опису, але це дещо знижує загальну зручність для користувача.

Оцінка якості відповідей проводилася за суб'єктивною шкалою - **"відмінно", "добре", "задовільно", "потребує доопрацювання"**. Результати:

- Відмінно — 15 відповідей;
- Добре — 4 відповіді;
- Задовільно — 1 відповідь;
- "Потребує доопрацювання" — 0 випадків.

Таким чином, якість генерації повідомлень можна оцінити як високу. Подальше вдосконалення може включати:

- автоматичне скорочення занадто довгих анонсів;
- переклад заголовків англійською (якщо джерело іноземне);
- динамічні шаблони відповідей залежно від категорії.

3.4. Аналіз результатів тестування та формулювання висновків

Аналіз таблиці результатів тестування показав, що новинний бот здебільшого правильно і стабільно виконує класифікацію новин відповідно до очікуваних категорій. Із 20 тестових взаємодій у 16 випадках класифікація була абсолютно точною. Ще у 4 випадках спостерігалось відхилення в бік суміжної категорії, що характерно для коротких або міждисциплінарних новин.

Наприклад, одна з новин у категорії «Кіно/музика» була класифікована як «Технології», ймовірно, через згадку технологічних аспектів у змісті (рядок 7). Аналогічна ситуація у випадках, коли новини зі сфери «Технології» були визначені як «Кіно/музика» або «Світ». Це свідчить про те, що модель

орієнтується на ключові терміни, але іноді не враховує загальну інтенцію тексту.

Таким чином, модель класифікації працює стабільно, хоча певне змішування суміжних тем залишається. Однак критичних помилок або віднесення до зовсім невідповідних категорій не зафіксовано, що свідчить про її загальну ефективність. Для підвищення точності доцільно або уточнити самі категорії (наприклад, об'єднати «Кіно/музика» з «Культурою»), або навчати модель з урахуванням подібних тематичних перетинів.

Також важливо відзначити, що всі категорії були протестовані по 5 взаємодій на кожному. Це забезпечує об'єктивність аналізу.

Висновки до розділу 3

Проведене тестування новинного бота дозволило здійснити всебічну оцінку точності класифікації новин та якості відповіді. За результатами 20 тестових взаємодій встановлено, що **точність класифікації складає 80%**, а в контексті близьких тем — до **90–95%**.

Помилки класифікації переважно пов'язані з тематичним перетином (наприклад, технологічні новини у сфері кіно), що є очікуваним у випадках коротких повідомлень. Жодної критичної або хаотичної класифікації не виявлено.

Крім того, всі відповіді бота були коректно сформовані, із зручним для користувача форматуванням, що свідчить про надійність модуля генерації повідомлень. Інтерфейс працював стабільно, і навіть при багаторазовому натисканні на одну й ту ж кнопку не виникало технічних збоїв.

Таким чином, новинний бот показав високу ефективність, що підтверджує доцільність його подальшого розвитку та масштабування. Виявлені відхилення в класифікації можуть бути усунені шляхом розширення навчального корпусу та використання глибших мовних моделей.

ВИСНОВКИ

У ході кваліфікаційної роботи було **проаналізовано** сучасні підходи до розробки новинних ботів, їх основні типи, функції та сфери застосування, а також ключові вимоги до їх реалізації. Детально розглянуто роль та можливості методів машинного навчання та штучного інтелекту, зокрема класифікації текстів та обробки природної мови (NLP), для автоматизації обробки новинного контенту. Проведено огляд та порівняльний аналіз існуючих новинних ботів, що використовують технології штучного інтелекту, таких як Reuters News Tracer, CNN Bot, NewsBot (Microsoft), Ukrainian News Bots та Botpress NewsBot, що дозволило визначити їхні переваги, недоліки та архітектурні особливості.

Виконано розробку та імплементацію інтелектуального новинного бота. Обґрунтовано вибір методів машинного навчання, а саме класифікатора Multinomial Naive Bayes у поєднанні з TF-IDF для тематичної класифікації новин, а також моделі GPT-4 для перекладу та генерації відповідей природною мовою. Розроблено модульну архітектуру бота, що включає окремі компоненти для збору новин (з використанням RSS-стрічок та парсингу), їх обробки та класифікації, формування та генерації відповідей, а також інтерактивний користувацький інтерфейс на базі Telegram. Здійснено практичну реалізацію кожного модуля, включаючи завантаження даних, навчання класифікаційної моделі, інтеграцію з API великих мовних моделей та налаштування взаємодії з користувачем через месенджер. Проведене тестування підтвердило функціональність та ефективність розробленої системи у вирішенні поставлених завдань.

Перспективи розвитку проєкту передбачають розширення функціоналу

бота за рахунок впровадження більш складних алгоритмів персоналізації новинного контенту, адаптації до індивідуальних вподобань користувачів та розширення джерел збору інформації. Планується подальше дослідження та інтеграція новітніх моделей глибокого навчання та трансформерних архітектур для підвищення точності класифікації, поліпшення якості машинного перекладу та забезпечення більш глибокої, контекстно-залежної діалогової взаємодії з ботом, що дозволить створити ще більш адаптивну та інтелектуальну систему.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Aiogram. **aiogram 3.0 Documentation**. — [Online]. Available: <https://docs.aiogram.dev/>
2. BotFather. **Telegram Bot API**. — Telegram, 2023. — [Online]. Available: <https://core.telegram.org/bots/api>
3. Brown, T. et al. (2020). **Language Models are Few-Shot Learners (GPT-3)** // arXiv:2005.14165. — [Online]. Available: <https://arxiv.org/abs/2005.14165>
4. Chollet, F. (2021). **Deep Learning with Python**. — 2nd ed. — Manning Publications. — 504 p.
5. Hugging Face. **Transformers Documentation**. — [Online]. Available: <https://huggingface.co/docs/transformers/>
6. Jurafsky, D., & Martin, J. H. (2023). **Speech and Language Processing**. — 3rd ed. — [Online]. Available: <https://web.stanford.edu/~jurafsky/slp3/>
7. Microsoft. **What is NLP?** — Azure AI Services. — [Online]. Available: <https://learn.microsoft.com/en-us/azure/cognitive-services/nlp/>
8. Newspaper3k. **Documentation**. — [Online]. Available: <https://newspaper.readthedocs.io/>
9. OpenAI. **GPT-4 Technical Report**. — OpenAI, 2023. — [Online]. Available: <https://openai.com/research/gpt-4>
10. Python Software Foundation. **Python Documentation**. — [Online]. Available: <https://docs.python.org/3/>
11. Russell, S., & Norvig, P. (2021). **Штучний інтелект: сучасний підхід** / пер. з англ. — 4-те вид. — Київ: Діалектика. — 1168 с.
12. Vaswani, A. et al. (2017). **Attention Is All You Need** // Proceedings of NeurIPS. — [Online]. Available: <https://arxiv.org/abs/1706.03762>
13. Баранов, П. О., Ситник, С. М. (2021). **Інтелектуальні чат-боти: огляд архітектур та технологій** // Наукові записки НУ "Острозька академія". Серія "Комп'ютерні науки". — №20. — С. 45–52.

14. Коваленко, І. В. (2020). **Чат-боти у цифровому суспільстві: класифікація, застосування, виклики** // Вісник НТУУ "КПІ". Серія: ІК. — №45. — С. 70–77.
15. Швидкий, Ю. М. (2022). **Машинне навчання у створенні новинних ботів** // Інформаційні технології і засоби навчання. — №2 (88). — С. 34–40.
16. Яскевич, Владислав Олександрович (2023) *JavaScriptбібліотека React Навчальний посібник для студентів спеціальності Комп'ютерні науки* Київський університет імені Бориса Грінченка, Україна, Київ. <https://elibrary.kubg.edu.ua/id/eprint/48587/>
17. В Машкіна, ВО Абрамов, ТІ Носенко, ЄВ Іваніченко. Навчально-методичний посібник до виконання і захисту кваліфікаційної роботи бакалавра спеціальностей 122 Комп'ютерні науки для здобувачів вищої освіти денної форми навчання, Київський столичний університет імені Бориса Грінченка. Київ, 2024.- 43 с.
18. Теоретичні та практичні аспекти використання математичних методів та інформаційних технологій в освіті й науці: моногр. / за заг. ред. О. Литвин. — К.: Київ. ун-т ім. Б. Грінченка, 2021. — 332 с.

Додатки

Додаток А. Код основної програми

```
from aiogram import Bot, Dispatcher, types
from aiogram.types import Message, ReplyKeyboardMarkup, KeyboardButton
from aiogram.utils import executor
from newspaper import Article
import feedparser
import random
import logging
from datetime import datetime
import joblib
from openai import OpenAI
import os

# Налаштування логування
logging.basicConfig(level=logging.INFO)
logger = logging.getLogger(__name__)

# Конфігурація
API_TOKEN = "7889578054:AAEHXuWqLd_q9CXn9qGKAYTFSGww9rmopcl"
OPENAI_API_KEY = "sk-proj-IvZRtwCii9q8L1pYOjejdi-7GJ6gULJRyFhmd5R5eufGUoRjp4fF2u2VVxTQNpewc_06ymuV_T3BlbkFJWSZfPv7F7qqeOqZDfnVU2cjb0OCSGJaRr9aD5_ydphLnk4AB7iTWTpzuqCCzV7XoLzBq3kzmka"

# Ініціалізація
bot = Bot(token=API_TOKEN)
dp = Dispatcher(bot)
client = OpenAI(api_key=OPENAI_API_KEY)
news_model = joblib.load("news_classifier.pkl")
user_histories = {}

# Джерела новин та RSS
rss_sources = {
    "ukraine_general": [
        "https://www.pravda.com.ua/rss/",
        "https://www.radiosvoboda.org/api/zmgppme$pm",
```

```

    "https://suspilne.media/rss/news.xml",
    "https://tsn.ua/rss",
    "https://www.unian.ua/static/rss/main.xml"
  ],
  "world": [
    "https://rss.dw.com/rdf/rss-en-all",
    "https://feeds.bbc.co.uk/news/world/rss.xml",
    "https://rss.cnn.com/rss/edition_world.rss",
    "https://www.aljazeera.com/xml/rss/all.xml"
  ],
  "sport": [
    "https://www.ua-football.com/rss/all.xml",
    "https://sport.ua/rss",
    "https://xsport.ua/rss/",
    "https://www.espn.com/espn/rss/news"
  ],
  "entertainment": [
    "https://life.nv.ua/rss/all.html",
    "https://kino-teatr.ua/uk/main/rss.rss",
    "https://www.billboard.com/feed/",
    "https://www.variety.com/feed/"
  ],
  "technology": [
    "https://ain.ua/feed/",
    "https://itc.ua/feed/",
    "https://www.theverge.com/rss/index.xml",
    "https://techcrunch.com/feed/"
  ]
}

```

для кнопки "Всі Джерела"

```

site_sources = {
  "ukraine_general": [
    "https://www.pravda.com.ua",
    "https://www.radiosvoboda.org",
    "https://suspilne.media",
    "https://tsn.ua",
    "https://www.unian.ua"
  ],

```

```

"world": [
    "https://www.bbc.com",
    "https://edition.cnn.com",
    "https://www.aljazeera.com",
    "https://www.dw.com",
    "https://www.reuters.com"
],
"sport": [
    "https://www.ua-football.com",
    "https://sport.ua",
    "https://xsport.ua",
    "https://www.espn.com",
    "https://www.skysports.com"
],
"entertainment": [
    "https://life.nv.ua",
    "https://kino-teatr.ua",
    "https://www.billboard.com",
    "https://www.variety.com",
    "https://playua.net"
],
"technology": [
    "https://ain.ua",
    "https://itc.ua",
    "https://www.theverge.com",
    "https://techcrunch.com",
    "https://dou.ua"
]
}
english_trigger_words = [
    "the", "is", "to", "of", "and", "in", "on", "with", "that", "for",
    "from", "by", "about", "this", "will", "you", "your", "their", "at", "as",
    "are", "be", "it", "was", "have", "has", "an", "or", "but", "we", "they", "more",
    "can", "new", "all", "not", "just", "if", "when", "up", "out", "what", "which"
]
def classify_news_title(title: str) -> str:
    try:
        return news_model.predict([title])[0]
    except Exception as e:

```

```

logger.error(f"Класифікація не вдалася: {e}")
return "невизначено"

# Перевірка, чи є англомовний текст
def is_english(text):
    words = text.lower().split()
    match_count = sum(1 for word in words if word in english_trigger_words)
    return match_count / max(len(words), 1) > 0.2

# Переклад тексту українською
def translate_to_ukrainian(text):
    try:
        response = client.chat.completions.create(
            model="gpt-4",
            messages=[
                {"role": "system", "content": "Ти перекладач з англійської на українську."},
                {"role": "user", "content": f"Переклади українською:\n{text}"}
            ],
            temperature=0.3,
            max_tokens=1000
        )
        return response.choices[0].message.content.strip()
    except Exception as e:
        logger.error(f"Помилка перекладу: {e}")
        return text

def get_categories_keyboard():
    keyboard = ReplyKeyboardMarkup(resize_keyboard=True)
    buttons = [
        KeyboardButton(text="🔍 Випадкова новина"),
        KeyboardButton(text="⚽ Спорт"),
        KeyboardButton(text="🎮 Кіно/Музика"),
        KeyboardButton(text="💻 Технології"),
        KeyboardButton(text="🌐 Світові новини"),
        KeyboardButton(text="📚 Всі Джерела")
    ]
    keyboard.add(*buttons[:2])

```

```

keyboard.add(*buttons[2:4])
keyboard.add(*buttons[4:6])
return keyboard

def get_category_name(key):
    names = {
        "ukraine_general": "Україна",
        "world": "Світ",
        "sport": "Спорт",
        "entertainment": "Кіно/Музика",
        "technology": "Технології"
    }
    return names.get(key, key)

# Отримання новини з перекладом
def get_rss_news(category=None):
    try:
        category = category or random.choice(list(rss_sources.keys()))
        feed_url = random.choice(rss_sources[category])
        feed = feedparser.parse(feed_url)
        if not feed.entries:
            return "Новини не знайдено."

        entry = random.choice(feed.entries)
        title = entry.title
        summary = entry.summary if hasattr(entry, 'summary') else ""

        # Переклад якщо потрібно
        if is_english(title):
            title = translate_to_ukrainian(title)
            summary = translate_to_ukrainian(summary)

        text = f"📰 *{title}*\n\n{summary[:1000]}"
        text += f"\n📊 Категорія (ML): {get_category_name(classify_news_title(title))}"
        text += f"\n📅 {datetime.now().strftime('%d.%m.%Y %H:%M%')}"
        text += f"\n[Джерело]({entry.link})"
        return text
    except Exception as e:

```

```

logger.error(f'Помилка при отриманні RSS новин: {e}')
return "Не вдалося завантажити новину."

@dp.message_handler(commands=['start', 'help'])
async def send_welcome(message: Message):
    await message.reply(
        "Вітаю! Я україномовний бот новин. Оберіть категорію або натисніть /news
для випадкової новини.",
        reply_markup=get_categories_keyboard())

@dp.message_handler(commands=['news'])
async def send_random_news(message: Message):
    await message.reply("Шукаю новину...")
    text = get_rss_news()
    await message.reply(text, parse_mode="Markdown",
disable_web_page_preview=False)

@dp.message_handler(lambda message: message.text in [
    "🔍 Випадкова новина", "⚽ Спорт", "🎮 Кіно/Музика",
    "💻 Технології", "🌍 Світові новини"])
async def category_news(message: Message):
    category_map = {
        "🔍 Випадкова новина": None,
        "⚽ Спорт": "sport",
        "🎮 Кіно/Музика": "entertainment",
        "💻 Технології": "technology",
        "🌍 Світові новини": "world"
    }
    category = category_map[message.text]
    await message.reply("Завантажую...")
    text = get_rss_news(category)
    await message.reply(text, parse_mode="Markdown",
disable_web_page_preview=False)

@dp.message_handler(lambda message: message.text == "📖 Всі Джерела")
async def all_sources(message: Message):
    response = "📖 *Список джерел новин:*\n\n"
    for cat, urls in site_sources.items():

```

```

        response += f"*{get_category_name(cat)}*:\n" + "\n".join(urls) + "\n\n"
        await message.reply(response, parse_mode="Markdown",
disable_web_page_preview=True)

```

```
@dp.message_handler()
```

```
async def ai_response(message: Message):
```

```
    user_id = message.from_user.id
```

```
    user_text = message.text
```

```
    # Пропускаємо системні кнопки
```

```
    if user_text in [
```

```
        " Випадкова новина", "⚽ Спорт", "🎮 Кіно/Музика",
```

```
        "💻 Технології", "🌐 Світові новини", "📖 Всі Джерела"
```

```
    ]:
```

```
        return
```

```
    history = user_histories.get(user_id, [])
```

```
    history.append({"role": "user", "content": user_text})
```

```
    await message.reply("💭 Думаю над відповіддю...")
```

```
    try:
```

```
        response = client.chat.completions.create(
```

```
            model="gpt-4",
```

```
            messages=[{"role": "system", "content": "Ти дружній україномовний асистент."}] + history,
```

```
            temperature=0.7,
```

```
            max_tokens=500
```

```
        )
```

```
        answer = response.choices[0].message.content.strip()
```

```
    except Exception as e:
```

```
        logger.error(f"Помилка OpenAI API: {e}")
```

```
        answer = "Вибачте, зараз не можу відповісти. Спробуйте пізніше."
```

```
    history.append({"role": "assistant", "content": answer})
```

```
    user_histories[user_id] = history[-10:] # Обрізаємо історію до останніх 10 повідомлень
```

```
        await message.reply(answer, parse_mode="Markdown",
reply_markup=get_categories_keyboard())
```

```
if __name__ == '__main__':
    logger.info("Запуск бота...")
    executor.start_polling(dp, skip_updates=True)
```

Додаток Б. Код моделі МН

```
import pandas as pd
```

```

from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import Pipeline
import joblib

# Завантаження очищеного CSV-файлу
df = pd.read_csv("news_rss_dataset.csv")

# Побудова моделі машинного навчання
model = Pipeline([
    ("tfidf", TfidfVectorizer()),
    ("clf", MultinomialNB())
])

# Навчання моделі
model.fit(df["title"], df["category"])

# Збереження моделі у .pkl файл
joblib.dump(model, "news_classifier.pkl")

print("✅ Модель навчено та збережено як news_classifier.pkl")

```

Додаток В. Код навчання моделі МН

```

import feedparser
import pandas as pd
from tqdm import tqdm

# RSS-джерела за категоріями
rss_sources = {
    "sport": [
        "https://www.ua-football.com/rss/all.xml",
        "https://sport.ua/rss",
        "https://xsport.ua/rss/"
    ],
    "technology": [

```

```

        "https://ain.ua/feed/",
        "https://itc.ua/feed/",
        "https://techcrunch.com/feed/"
    ],
    "entertainment": [
        "https://life.nv.ua/rss/all.html",
        "https://kino-teatr.ua/uk/main/rss.rss",
        "https://variety.com/feed/",
        "https://www.billboard.com/feed/",
        "https://feeds.feedburner.com/thr/news",
        "https://feeds.ign.com/ign/all",
        "https://www.comingsoon.net/feed"
    ]
},
    "world": [
        "https://rss.cnn.com/rss/edition_world.rss",
        "https://feeds.bbc.co.uk/news/world/rss.xml",
        "https://www.aljazeera.com/xml/rss/all.xml"
    ],
    "economy": [
        "https://biz.nv.ua/rss/all.html",
        "https://www.epravda.com.ua/rss/"
    ]
}

```

Збір новин із RSS

```
entries = []
```

```
print("🔍 Збір новин з RSS...")
```

```
for category, urls in tqdm(rss_sources.items()):
```

```
    for url in urls:
```

```
        feed = feedparser.parse(url)
```

```
        for entry in feed.entries:
```

```
            title = entry.title.strip()
```

```
            if title and len(title) > 15:
```

```
                entries.append({"title": title, "category": category})
```

Створення DataFrame

```
print(f"📁 Зібрано {len(entries)} заголовків. Очищення та збереження...")
df = pd.DataFrame(entries).drop_duplicates(subset=["title"])
df = df.sample(frac=1).reset_index(drop=True).head(1000)

# Збереження
df.to_csv("news_rss_dataset.csv", index=False)
print("✅ Файл збережено як news_rss_dataset.csv")
```

Додаток Г. Таблиця 1 Таблиця-порівняння чат-ботів

№	Назва бота	Платформа	Основна функція	Використання ШІ/ML	NLP/Діалог	Мова/Локалізація
1	Reuters News Tracer	Веб / редактори	Виявлення подій, класифікація/фільтрація	Так	Ні	Англійська
2	CNN Bot	Facebook Messenger	Персоналізовані новини, фільтрація	Так	Частково	Англійська
3	NewsBot (Microsoft)	Веб / Messenger	Діалог, агрегація, аналітика	Так	Так	Мультимовний
4	Ukrainian News Bots	Telegram	Категоризація, переклад, локальні новини	Частково	Частково	Українська/багатомовна
5	Botpress NewsBot	Botpress (Web, TG)	Діалоги, сценарії, ML - інтеграція	Так	Так	Залежить від налаштувань

Додаток Д. Таблиця 2 Точність класифікацій новин

№	Категорія (натиснута кнопка)	Категорія (МН)
1	Спорт	Спорт
2	Спорт	Спорт
3	Спорт	Спорт
4	Спорт	Спорт
5	Спорт	Спорт
6	Кіно/музика	Кіно/Музика
7	Кіно/музика	Технології
8	Кіно/музика	Кіно/Музика
9	Кіно/музика	Світ
10	Кіно/музика	Кіно/Музика
11	Технології	Технології
12	Технології	Кіно/Музика
13	Технології	Світ
14	Технології	Технології
15	Технології	Технології
16	Світові новини	Світ
17	Світові новини	Кіно/Музика

№	Категорія (натиснута кнопка)	Категорія (МН)
18	Світові новини	Світ
19	Світові новини	Світ
20	Світові новини	Світ