UDC 004.623

ІНТЕЛЕКТУАЛЬНА СИСТЕМА АНАЛІЗУ ТА ДЕТЕКЦІЇ ТЕКСТОВОГО БОТ-КОНТЕНТУ ВЕЛИКОГО ОБСЯГУ У СОЦІАЛЬНИХ МЕРЕЖАХ

Ph.D. М. Рудніченко ORCID: 0000-0002-7343-8076
Національний університет «Одеська Політехніка», Україна
E-mail: nickolay.rud@gmail.com
Ph.D. Н. Шибаєва ORCID: 0000-0002-7869-9953
Національний університет «Одеська Політехніка», Україна
E-mail: nati.sh@gmail.com
Ph.D. Т. Отрадська ORCID: 0000-0002-5808-5647
Одеський коледж комп'ютерних технологій «СЕРВЕР», Україна
E-mail: tv_61@ukr.net
Д. Шведов ORCID: 0009-0002-4823-8782
Національний університет «Одеська Політехніка», Україна
E-mail: frumle@ukr.net
Ph.D. І. Шпінарева ORCID: 0000-0001-9208-4923
Національний університет «Одеська Політехніка», Україна
E-mail: iryna.shpinareva@onu.edu.ua
Dr.Sci. І. Петров ORCID: 0000-0002-8740-6198
Національний університет «Одеська Політехніка», Україна
E-mail: firmn@gmail.com

Abstract. *У роботі розглядаються різні аспекти розробки інтелектуальної системи аналізу та обробки великих обсягів неоднорідних текстів природною мовою для завдання виявлення ботів у соціальних мережах із використанням методів глибокого перенесення навчання, зокрема великих мовних моделей. Наведено детальний аналіз специфічних характеристик та ключових аспектів структурування, обробки та аналізу текстового контенту, обґрунтовано актуальність проблеми та проведено огляд існуючих підходів у науковій літературі. У роботі підкреслено переваги та потенційні можливості використання штучних нейронних мереж і машинного навчання для автоматизації аналізу текстів користувачів соціальних мереж. Надано опис набору даних, обраного для дослідження, обґрунтовано вибір мовних моделей на основі штучних нейронних мереж та пояснено використання перенесення навчання для адаптації цих моделей до завдання виявлення ботів. Описано технічні засоби та сервіси, що використовуються для реалізації функціоналу розробленого веб-додатку, а також створено об'єктно-орієнтовані моделі системи за допомогою UML, включаючи діаграми прецедентів та компонентів. Окреслено функціональні можливості програмного забезпечення, прототипи сторінок та графічний інтерфейс користувача. У роботі представлені результати експериментальних досліджень обраних мовних*

моделей на розширеному наборі даних, перевірених у режимах як з текстовими поясненнями, так і без них. Проаналізовано продуктивність адаптованих нейронних моделей на певному етапі, визначено особливості їхньої роботи та запропоновано перспективні напрями для подальших досліджень і розробок для вирішення виявлених проблем.

Ключові слова: *інтелектуальний аналіз даних, класифікація тексту, соціальні мережі, аналіз тексту, обробка природної мови, розробка програмного забезпечення, виявлення ботів, великі дані*

1. Вступ

У сучасному інформаційному суспільстві Інтернет можна назвати невід'ємною частиною бізнесу, що дозволяє будь-якій компанії здійснювати ділові комунікації з такими цільовими групами, як клієнти, торгові посередники (канали збуту), PR-сфера, постачальники, конкуренти, чинні та потенційні співробітники [1].

Під час проведення подібних комунікацій генеруються, обробляються і зберігаються великі обсяги різноманітних даних, зокрема мультимедійні файли (зображення, відео), і навіть який завжди чітко структурований текстовий контент [2]. В даному контексті слід зазначити, що однією з головних тенденцій розвитку Інтернету останніх років є стрімке зростання популярності соціальних мереж (SN), які все частіше та активніше використовуються в маркетингових цілях, зокрема для просування того чи іншого товару, послуги, експерта, лідера думок , програмних додатків та сервісів та ін.

У цих умовах використання SN як джерела отримання даних та формування інформаційної бази по клієнтам є доцільною та важливою [3].

Для сучасних SN можна виділити такі характерні ефекти та властивості, які важливо враховувати при їх використанні для вирішення завдань бізнесу:

– наявність власних думок користувачів, що беруть участь у системі; зміна думок членів SN під впливом інших;

– різна значущість (пріоритетність або вага) думок одних користувачів для інших через їх рівень експертизи;

– змінюється ступінь схильності членів SN до впливу між собою;

– наявність непрямого впливу чи залежностей між користувачами у всьому ланцюзі наявних соціальних контактів; існування експертів, тобто. «лідерів думок» у певній тематиці;

– наявність деякого порога чутливості до зміни у лексиці оточуючих;

– локалізація створених груп за ознаками (за інтересами, з близькими думками, об'єднаних за соціальними чи гендерними ознаками);

– облік факторів "соціальної кореляції";

– існування зовнішніх факторів впливу та сторонніх агентів (ЗМІ, продавці чи виробники);

– вплив SN на динаміку думок у віртуальному співтоваристві;

– можливість утворення коаліцій чи команд, груп за інтересами;

– ігрову або інтерактивну взаємодію користувачів в інтерактивному режимі [4-6].

При цьому слід зазначити, що в даний час все більшої актуальності набувають проблеми захисту інформації та протидії інформаційним загрозам у різних системах обміну даними, у тому числі і в SN, контент у яких формується різними способами та методами, від ручного написання тематичних текстів до їх синтетичної генерації на основі застосування різних інтелектуальних технологій та засобів [7]. Фактично на практиці часто зустрічаються ситуації, коли потрібно розпочати листування з іншими користувачами SN з метою проведення консалтингу, обміну думками або оцінки характеру та якості контенту, що публікується [8].

Через те, що наведені вище процеси не можуть бути повноцінно автоматизовані і часто виконуються вручну, внаслідок чого є ресурсомісткими і дорогими для бізнесу в таких випадках потрібно гарантувати коректність інформації, що отримується, її цільовий характер і мінімізувати ризики отримання некоректних даних.

У зв'язку з цим актуальним завданням є ідентифікація та виявлення за прямими і непрямими поведінковими, лінгвістичними та семантичними ознаками автоматизованих програм (ботів), що ховаються під профілями користувача, що здійснюють публікацію недостовірних та свідомо неправдивих відомостей, що прагнуть обманним шляхом отримати особисту номери телефонів, скани документів, номери платіжних реквізитів, кредитних та дебетових карток та ін.), а також сприяють дестабілізації настроїв у суспільстві SN, що призводить до негативних наслідків для бізнесу при просуванні товарів та послуг [9].

Бот або віртуальний помічник (ВП) у широкому сенсі є спеціалізованим програмним забезпеченням, здатним здійснювати симуляцію дій реального користувача, зокрема генерувати текстовий контент [10]. Зокрема, у разі управління обліковим записом користувача за допомогою застосування такого ПЗ в автоматизованому режимі вважається, що даний процес реалізується ботом. У випадку, коли обліковий запис керується частково програмою та частково людиною, застосовується термін «кіборг». На практиці розрізняють прості боти, що діють директивно та виконують набір чітко заданих команд, а також за допомогою функцій самонавчання [11].

Боти мають ряд істотних переваг у порівнянні зі звичайними обліковими програмними додатками, наприклад, їх простіше встановити, не потрібно займати для зберігання даних пам'ять пристрою, посилання на отримання доступу до бота легко поширювати. Фактична зручність від використання ботів полягає у підтримці можливостей доставки повідомлень у зручні для отримання користувачем місця, наприклад, особисті повідомлення в SN, месенджерах.

ВП у ряді випадків застосовуються в діалогових системах для різних прикладних цілей, у тому числі для обслуговування клієнтів, агрегації даних та інформації. Боти уможливлюють спілкування компаній із замовниками в

індивідуальному інтерактивному режимі, що не вимагає залучення до цього процесу співробітників.

Боти можуть бути адаптовані для завдань автоматизації рекламування або надання облікових сервісів, від замовлення квитків на заходи, букінгу та бронювання місць у готелях, виконання порівняльного аналізу товарів та послуг за заданими критеріями. Можливим аспектом їх використання може бути автоматизація виконання різних рутинних операцій з великою кількістю клієнтів у банківському секторі, торгівлі [8].

Боти починають застосовуватися і в рамках державного сектору з метою реєстрації звернень громадян до міських служб, проведення обробки запитів по комунальному господарству, що надходять від людей, оплаті рахунків та відправці нагадування про необхідність внесення показань лічильників.

В даний час боти в SN класифікуються за різними ознаками та типами (рис.1.6), розглянемо найбільш популярні їх типи, до яких належать:

1. Соціальні боти – це спеціалізовані програми, використовувані насамперед для імітації процесу живої поведінки людини у SN. Тобто. соціальні роботи наслідують людську поведінку для акцентування уваги користувачів на заданій інформації. На практиці подібні роботи використовуються як для позитивних завдань (консультування користувачів, підтримка у проведенні опитувань, публікація контенту за розкладом, просування брендів або конструктивних ідей у SN, привернення уваги користувачів до соціальних проблем чи інших тематиків), так і для негативних завдань (виявлення та розкрадання персональних даних користувачів шляхом поширення недостовірної інформації у процесі імітації легітимних користувачів). Політики можуть використовувати подібний тип ботів для спілкування зі своєю цільовою аудиторією та отримання зворотного зв'язку, компанії застосовують їх як віртуальні агенти з обслуговування клієнтів.

2. Технічні роботи – різновид програми, що застосовуються для виконання одноманітних дій в SN, як правило з метою підвищення числа позитивних відгуків (накручування лайків), підвищення рівня ранжування інформації в стрічці новин, і додавання простих коментарів з метою подальшого репоста. Даний тип роботів реалізують функцію соціалізації для інших користувачів або роботів в SN шляхом додавання їх у відповідні тематичні списки з метою підвищення рівня довіри до них у активної спільноти

3. Бойові боти характеризуються призначенням щодо відкладених (спланованих у часі за розкладом) різних активних дій, пов'язаних з обробкою облікових записів користувачів SN або публікацією цільового контенту. Як правило, дані боти використовуються для негативних завдань, вони можуть бути тривало не активними, активізуюся в заданий момент і отримуючи доступ до персональних даних користувача в момент початку інформаційної операції використовуються, наприклад, для публікації негативних відгуків або коментарів [10].

4. Тролі представляють найбільш агресивний та негативний тип ботів у SN. Даний тип ґрунтується на застосуванні різних семантичних та лексичних наборів даних, що є деякою подобою словників і містять текстові набори

різних ключових слів за заданою тематикою. За словниками роботи знаходять необхідні цільові публікації в постах або стрічках SN, публікують ворожі та образливі коментарі користувача, тим самим провокуючи не конструктивні та асоціальні суперечки між членами віртуального співтовариства, формуючи негативний інформаційний фон та поширюючи деструктивний інформаційний.

5. Дезінформатори є окремим гібридним видом ботів-тролів та соціальних ботів для виконання негативних завдань з імітації діяльності користувачів у SN, входять у довіру співтовариства, після чого плавно публікують заданий шкідливий або хибний контент для інформаційного впливу на користувачів. У випадках їх успішної діяльності сформована дезінформація лавиноподібно поширюються як у рамках SN, так і поза нею, зокрема у ЗМІ.

6. Боти-спамери застосовуються в SN для поширення контенту не несе користі, зокрема це може бути публікація застарілих та не актуальних відомостей. Метою цього типу шкідливих ботів є ускладнення сприйняття користувачами публікованого контенту, внесення шуму чи засмічення хеш-тегів, тобто. спамери призначені виключно для виконання зловмисних дій. Популярними діями даного типу роботів є поширення інформаційного чи персонального контенту серед користувачів без дозволу правовласниками.

7. Впливові боти - це тип ботів, які здійснюють інформаційні дискусії щодо обраних тенденцій або тематик у розділах SN для просування та популяризації заданих тем. Частково їх принцип роботи ґрунтується на ботах дезінформаторах, при цьому їх складніше виявити. Цей тип роботів зазвичай генерують повідомлення шляхом повторного використання чужого контенту з незначним рерайтом вмісту або створюють авторські повідомлення за допомогою певного семантичного набору правил. Оскільки впливові облікові записи мають одну зі своїх цілей щодо поширення контенту на максимальну кількість людей, тому вони намагаються надіслати найбільшу кількість запитів у друзі до розповсюдження контенту. Вплив конкретного користувача в SN залежить від його рівня популярності та довіри в мережі, іноді цей критерій оцінюється як кількість вхідних запитів або отриманих повідомлень. Основне завдання подібних ботів полягає у зміні думки або відношення користувача з конкретної тематики або продукту [11].

Останнім часом, починаючи з фази появи COVID-19, ситуація з проявом негативної активності з боку ботів значно загострилася. Ризики використання ботів пов'язані з різними аспектами інформаційної безпеки, що найчастіше зустрічаються такі негативні аспекти їх використання на практиці:

1. Використання не ефективно налаштованих та функціонуючих ботів у службах підтримки клієнтів може призвести до втрат у бізнесі та подальшого банкрутства компанії. Слід зазначити, що рівень довіри до компанії знижується у разі, коли в роботі робота спостерігається надмірна кількість безглуздих питань, відсутність емоційного співчуття та конкретних відповідей на питання, а також неможливість підтримки процесу вирішення складних складових проблем. Все це призводить до того, що клієнти почуваються обдуреними.

2. Використання ботів для завдань збору даних (парсингу) із сторонніх ресурсів. Загрози від використання роботів даному ракурсі полягають у

крадіжці унікального контенту з ресурсу з метою подальшої його публікації без посилань на джерело. Це призводить до того, що рівень авторитетності ресурсу-джерела в пошуковій видачі знижується, призводячи до скорочення цільової аудиторії, рівня продажу та обсягів рекламних доходів. У тому числі збирання даних використовується для автоматизації відстеження цін на товари у конкурентів для подальшого демпінгу та відсікання клієнтської бази або для перепродажу товарів з націнкою. Можливі ризики створення хибних замовлень з метою завантаження логістичних ресурсів збільшення витрат компаній конкурентів і забезпечення тимчасової недоступності товарів замовлення іншими користувачами [8].

3. Накручування показників. Боти вносять шумові дані в статистику та аналітику бізнес-завдань компанії, впливаючи на показники конверсії, кількість лідів, параметри вирви продажів, що веде до нераціональних рішень та зайвих маркетингових витрат. У контексті SN некоректні дані можуть бути внесені до спільноти або групи під час проведення онлайн-голосувань, де використовувані боти можуть штучно завищувати різні показники для просування своїх цілей чи людей.

4. Автоматизація проведення DDoS-атак (класу "відмова в обслуговуванні"), які сфокусовані на вплив за обчислювальними системами (веб-серверами та серверами додатків, іноді це також можуть бути канали зв'язку) для створення ситуації, в якій дані системи та сервіси стають недоступним користувачам. Технічно DDoS-атаки виражаються у формі одночасної (паралельної) відправки на адресу певного веб-сайту або сервісу великої кількості запитів на читання даних з безлічі розподілених комп'ютерів у мережі Інтернет. Процес впливу зумовлений тим, що велика кількість (десятки і сотні тисяч) запитів з різних комп'ютерів (керованих ботами) за адресою певного сервера (сервісу) призводить до фізичної неможливості їх оперативної обробки, тобто. може бути переповнені обсяги виділеної для обробки даних пам'яті або смуга пропускання каналу зв'язку не зможе забезпечити проходження запитів на сервер

І в першому, і в другому випадках, користувачі не зможуть отримати доступ до такого сервісу або до інших веб-сайтів (на вкладених доменах, зокрема) через звернення через канал зв'язку. Метою проведення DDoS-атак є отримання матеріальної вигоди за допомогою виконання здирництва за припинення таких атак, або досягнення необхідних політичних інтересів.

Найбільш активно використовуваним методом ефективних DDoS-атак є метод створення ботнетів, які є заражені вірусним ПЗ комп'ютери, що містять у своєму складі відповідні програмні закладки для реалізації потрібної реакції.

У більш широкому значенні ботнетом називається мережа автоматизованих комп'ютерних роботів. Кожному роботі в цій мережі надається свій набір цільових завдань, які повинні виконуватися автоматично за певними подіями або умовами. У зв'язку з тим, що мережа ботів складається з їх наборів – контролер (керівна ланка системи) може динамічно виконувати різні шкідливі завдання, у тому числі розповсюдження небажаного рекламного, хибного чи вірусного контенту, надсилання запитів користувачам із соціального графа

(переліку друзів) користувачів SN з метою отримання коштів та заклику до інших дій [12].

2. Постановка проблеми та аналіз існуючих підходів та публікацій

Останні досягнення в мовних моделях можна пояснити переважно методами глибокого навчання, прогресом у нейронних архітектурах, таких як трансформери, розширеними обчислювальними можливостями та доступністю навчальних даних, отриманих з Інтернету. Ці розробки призвели до революційної трансформації, дозволивши створити LLM, здатні наближати продуктивність людського рівня за певними тестами оцінки [9, 16].

LLM, особливо попередньо підготовлені види даних моделей, згідно з рядом досліджень [17] здатні надати широкі можливості для розуміння, аналізу, оцінки та генерації текстового контенту широкому діапазоні завдань.

У зв'язку з цим попит на LLM виріс, у тому числі через зростаючу потребу в машинах для виконання складних мовних завдань, таких як переклад, резюмування, пошук інформації та розмовна взаємодія. LLM досягають цієї майстерності шляхом самостійного навчання на великих наборах текстових даних [18].

Після точного налаштування (тюнінгу) виконання завдань з аналізу різнорідних текстів великих обсягів LLM демонструють істотне підвищення продуктивності, часом [19] перевищуючи ефективність моделей, навчених повністю з нуля.

Ці риси мовних моделей сприяють застосуванню LLM при їх навчанні на великих наборах даних, що дозволяє відзначити той факт, що масштабування розмірів самих моделей і обсягів використовуваних для навчання і тестування наборів даних призводить до подальшого вдосконалення їх узагальнюючої здатності.

LLM також починають часто використовуватись у бізнесі. Наприклад LLM можливо використовувати для менеджменту бізнес-процесів, зокрема виконання таких процесів: видобуток імперативних моделей процесів з текстових описів, видобуток декларативних моделей процесів з текстових описів, та оцінка придатності завдань процесу з текстових описів для роботизованої автоматизації процесів. Вони показали, що без масштабної конфігурації чи prompt-engineering LLM працюють порівняно або краще, ніж існуючі рішення. Це дослідження є аргументом на користь того, що має сенс порівнювати різня LLM задля обрання найкращих з них для бізнес-задач.

LLM часто базуються на архітектурі глибинних нейронних мереж, які застосовують архітектуру трансформерів. Трансформери - це клас моделей глибокого навчання для роботи з послідовностями чи множинами даних, ці моделі засновані на механізмі самоуваги. Цей механізм дозволяє моделі акцентувати увагу на конкретних елементах послідовності залежно від інших елементів та краще розуміти зв'язки між словами. Вони ефективним чином вловлюють складні контекстуальні зв’язки в тексті, що робить їх основою

багатьох найсучасніших програм NLP. Їхня архітектура дозволяє виконувати паралельну обробку, що робить їх високоефективними та масштабованими для різних завдань, окрім обробки мови, таких як розпізнавання зображень і навчання з підкріпленням. Важливими аспектами трансформерів є [9]:

1. Механізм самоуважності. Цей динамічний механізм дає змогу моделі фіксувати довгострокові залежності та асоціації між словами в послідовності.

2. Увага кількох голов. Замість того, щоб покладатися на єдиний механізм уваги, трансформери використовують кілька "голів уваги", що працюють паралельно. Кожна головка спеціалізується на фіксуванні різних взаємозв'язків у даних, а їхні результати об'єднуються, щоб охопити широкий спектр інформації.

3. Складені шарів. Трансформери зазвичай складаються з кількох ідентичних слоїв, складених послідовно. Кожен шар удосконалює представлення, отримані від його попереднього, дозволяючи моделі охоплювати дедалі складніші та абстрактні характеристики.

4. Вихідний рівень у завданнях класифікації вихідний рівень генерує ймовірності класу, тоді як у завданнях генерації послідовності він послідовно створює вихідну послідовність.

5. Навчання: Трансформери навчаються за допомогою алгоритмів зворотного поширення та оптимізації, таких як ADAM. Специфічні для завдання функції втрат, такі як перехресна ентропія для класифікації, зведені до мінімуму під час навчання. Попереднє навчання великим мовним корпусам, як це видно в таких моделях, як BERT і GPT, стало ключовим фактором їх успіху.

Ця комбінація характеристик обумовлює успіх LLM на основі трансформерів, але при використанні даного механізму існують також і недоліки. Деякими з них є наступні [10]:

1. Обчислювальна складність. Навчання та використання трансформерів може бути витратним процесом, особливо для великих моделей, таких як GPT-3. Це може бути основною перешкодою для їх розгортання в реальних програмах, особливо для невеликих організацій і окремих осіб, які можуть не мати доступу до необхідних потужних обчислювальних ресурсів.

2. Перенавчання. Моделі трансформерів можуть бути легко перенавченні, особливо якщо дані невеликі або недостатньо різноманітні. Це може призвести до низької продуктивності узагальнення раніше невідомих даних. Це може відображатися у значній зміні результату роботи трансформера при не значній зміні опрацьованого речення без зміни його сенсу.

3. Довгострокові залежності. У моделей трансформерів можуть виникнути труднощі з моделюванням довгострокових залежностей у послідовностях, особливо в ситуаціях, коли залежності охоплюють кілька токенів. Це також залежить від методу позиційного енкодування.

4. Прихильність уваги. Моделі даного типу покладаються на механізми уваги, щоб визначити, які частини вхідної послідовності є найбільш релевантними. Однак іноді ці механізми можуть бути прихильні до деяких значень більш за інші, що призводить до неоптимальних результатів.

5. Можливість інтерпретації. Такі моделі можуть бути складними для інтерпретації та розуміння, оскільки вони не мають чітких проміжних представлень, які можна інтерпретувати, як ті, що виробляються деякими іншими архітектурами нейронних мереж.

Також слід відвести увагу на те, що великі мовні мережі за своєю архітектурою не моделюють чітких логічних зв'язків у реченні. Механізм самоуваги надає реальні числа як позначення уваги до елементів речення, що є механізмом нечіткої логіки, та не завжди буде ідеальним методом моделювання мови.

Аналізуючи характеристики LLM, та недоліки їх найпопулярнішої архітектури, зроблено висновок, що велика складність імплементацій трансформерів, а також характеристики та недоліки найпопулярнішої архітектури (трансформери), обгрунтовують створення платформи для експериментування над та порівняння не тільки існуючих архітектурних рішень LLM, а що й забезпечує можливість експериментувати над іншими майбутніми архітектурами BMM, та експериментувати з використанням майбутніх допоміжних методик задля покращення результатів роботи LLM. Слід зазначити, що згідно з рядом досліджень [20] на якість результатів LLM впливає не лише подання прикладів виконання завдання у запитів, але й те, як саме завдання було описано природною мовою в запиті.

Важливою частиною роботи з LLM є інжиніринг запитів (фрагменти текстових запитів, що надсилаються на вхід моделі, які формалізують завдання, яке LLM має виконати, з урахуванням додаткових правил, підказок, прикладів та змістового контексту) для підвищення ефективності та точності їх використання. Даний процес, на думку авторів [21], заснований на послідовному виконанні процедур зміни та оптимізації вхідних запитів для підвищення цільового результату, згенерованого LLM для прикладних завдань.

Важливим аспектом у цьому випадку є те, що підсумкова якість роботи моделі може значно змінюватись в залежності від того, як саме був сформований запит, навіть у тому випадку, коли два різні запити мають однакову сутність та мету, але різні процедури формування.

Як результат роботи подібної моделі ANN можна сформувати широкий розподіл ймовірностей можливих токенів, що є продовженням текстових речень. Вибір кінцевого токена для моделі часто визначається за допомогою виконання етапів вибірки даних та її тюнінгу, у цьому випадку важливу роль відіграє підбір значень гіперпараметрів, які можуть впливати на компроміс між різноманітністю та точністю згенерованого тексту [22].

Все це може бути корисним при оцінці текстових постів користувачів в SN для аналізу їх профілів щодо наявності аномальної поведінки і виявлення ботів.

Як результат роботи LLM отримується розподіл ймовірностей можливих токенів - продовжень речень. Вибір кінцевого токена часто визначається за допомогою процесу вибірки, де гіперпараметри відіграють ключову роль. Ці гіперпараметри можуть впливати на компроміс між різноманітністю та точністю згенерованого тексту [21].

Гіперпараметри, такі як "temperature", використовуються для контролю рівня випадковості у виводі LLM. Вище значення температури призводить до більш різноманітних, але потенційно менш зв'язних відповідей, оскільки модель досліджує ширший діапазон можливостей. І навпаки, нижче значення температури робить вихід більш детермінованим, сприяючи точності та узгодженості за рахунок різноманітності. Це створює додатковий аспект у порівнянні LLM, бо різні комбінації гіпермараметрів можуть призвести до різної якості результатів роботи LLM [22]. Це особливо стосується пропрієтарних моделей, бо компанії які їх надають частіше не дозволяють переглянути отриманий розподіл ймовірностей токенів, тому для оцінки якості моделі з різними гіперпараметрами знадобиться проводити експериментування з різними параметрами, на відміну від можливого аналізу отриманого розподілу ймовірностей при використанні відкритих LLM. Слід зазначити, що одним із ключових недоліків поширених типів архітектур LLM, наприклад, трансформерів, є неможливість моделювати чітку логіку виконання запиту та схильність моделей до фактичних помилок на великих текстових промптах [11,17,19]. Для боротьби з подібними проблемами активно розробляються та досліджуються різні методи підвищення точності та якості роботи LLM під різні змістові контексти.

Прикладом підходу, що використовується, наприклад, є підключення реляційних або не реляційних баз даних як джерела актуальної інформації та символічної пам'яті, що спрощує моделям процес обробки даних.

У цьому випадку, поєднуючи метод ланцюжка знань та БД, можна надати ВММ можливість доступу до фактичної та символічної інформації, отриманої або збереженої за потребою [17].

Таким чином, аналізуючи існуючі праці з дослідження даної тематики, незважаючи на виявлені складності використання LMM та їх недоліки, слід відзначити актуальність та доцільність розробки та використання таких моделей для завдань аналізу текстових даних, зокрема, у контексті виявлення ботів у SN.

Метою роботи є розробка та дослідження інтелектуальної системи аналізу та детекції текстового бот-контенту великого обсягу у соціальних мережах на базі застосування глибинного навчання та LLM підходу.

3. Розробка концепції проекту

Для реалізації та застосування функціоналу моделей ANN у рамках аналізованої проблематики необхідно знайти або створити набори тестових даних значного обсягу. Як основу вирішено використовувати існуючі фрагменти наборів даних, що знаходяться у відкритому доступі, провівши їх попередню обробку, очищення, а також агрегувати ряд адаптованих вибірок для надання даних більшого балансування та різноманітності. Для завдання виявлення ботів у SN, яку доцільно звести до завдання класифікації використано набір даних “PAN19 Author Profiling” [23]. Цей набір даних був

створений з метою допомогти визначати онлайн-ботів користувачів. Набір даних включає по 100 публікацій різних користувачів у соціальній мережі Twitter, а також індикатор того, чи є даний користувач ботом. Набір даних є збалансованим. Для оцінки адекватності та точності LLM за цим набором даних доцільним є використання метрик бінарної класифікації, такі як accuracy, recall, precision, f1. Для проведення експериментів над популярними та новими LLM необхідно забезпечити доступ до їхнього функціоналу за допомогою підключення доступних API. Провівши аналіз доступних варіантів встановлено, що:

– Для пропрієтарних моделей доступ надається найчастіше завдяки спеціально розробленим для них API під завдання генерації тексту.

– Для open-source моделей можна отримати доступ завдяки публічним репозиторіям та реалізованим драйверам підтримки їхнього залучення в роботу.

– Для самостійної розробки мовних моделей необхідно створити свій або використовувати існуючий фреймворк тренування та активації LLM.

В рамках цього дослідження прийнято рішення використовувати такі моделі LLM (адаптувавши їх під наше завдання): GPT2, Bloomz-1b1 та Mistral-7B. GPT2. У порівнянні з найновішими моделями GPT2 має значно менше параметрів та значно меншу здатність розуміти текст. Але через невеликий розмір моделі було вирішено використовувати GPT2 як базовий "нульовий" рівень якості при порівнянні з іншими моделями, тому що її результат роботи можна швидко обчислити, що сприяє реалізації концепції порівняльного тестування моделей. Bloomz-1b1.

Це open-source LLM приймає близько 1.1 мільярд параметрів, що відносно мало в порівнянні з іншими моделями. Це зменшує її потенціал до розуміння тексту, але її використання дозволить виміряти наскільки гнучкими завдання виявлення ботів в SN можуть бути LLM при відносно малій кількості параметрів. Також це дозволить проводити локальні експерименти відносно швидко.

Дана модель спочатку навчена для аналізу семантичних інструкцій у тексті, що доводить її доцільність при аналізі текстових постів розмовної стилістики. Mistral-7B. Модель з відкритим кодом розроблена MistralAI. модель, розроблена на вирішення завдань NLP з високим рівнем продуктивності. На думку авторів [24], Mistral 7B перевершує Llama 2 13B за всіма оціночними показниками, модель використовує увагу до згрупованих запитів для більш швидкої генерації, у поєднанні з увагою до ковзних вікон для ефективної обробки послідовностей довільної довжини зі зниженою швидкістю генерації. Використання даної моделі дозволить більш репрезентувати сферу розробки LLM з відкритим кодом, в даному випадку представляється модель більшого розміру, ніж bloomz-1b1 і з можливістю вказувати інструкції. Для адаптації моделей даної задачі пропонується використовувати концепцію індуктивного типу transfer learning (TL) з елементами кросмодальності [11, 17, 25].

Якщо $f_{w,s} : X \to Y$ буде попередньо навченою моделлю на вихідному наборі даних D_s де $w_s \in \Re^D$ позначає D-вимірний ваговий вектор попередньо підготовленої LLM. Враховуючи цільовий набір даних D_t, метод тонкого налаштування мінімізує стандартний негативний логарифм правдоподібності $L_t(w) = \sum_{i=1}^{N_t} \log\ p_w(\frac{y_t^i}{x_t^i})$ за допомогою стохастичного градієнтного спуску $w(t+1) = w(t) - \eta \nabla_w L_t(w), w_0 = w_s$, де η це розмір кроку і $\nabla_w L_t(w)$ позначає стохастичну оцінку градієнта втрат з використанням міні-пакета даних.

Таким чином, точне налаштування є оцінкою максимальної правдоподібності, на якій зосереджено логарифмічний пріоритет w_s.

Використовуючи наведені вище передбачені моделі, ми скорочуємо час навчання тренуючи тільки останній шар моделей зі значно меншою кількістю змінних. Це пов'язано з тим, що якщо не "заморозити" змінні попередньої моделі, то в процесі навчання на новому наборі даних значення змінних будуть змінюватися (останній шар буде заповнений випадковими значеннями), у зв'язку з цим моделі можуть допускати великі помилки при аналізі тексту, що, у свою чергу, спричинить сильні зміни вихідних ваг у передбачуваній моделі.

Перевагою доступу до вибраних моделей LLM за допомогою вибраного API є підтримка використання обчислювальних можливостей компанії, що надає доступ до LLM, але недоліком цього є різні URL-адреси та формати запитів до API залежно від політики та обмежень, що накладаються компанії, що надає доступ до LLM.

Це ускладнює проведення досліджень, т.к. необхідно реалізувати методи здійснення різних форматів запиту.

Для вирішення цієї проблеми було вирішено використати сервіс OpenRouter. Цей сервіс дозволяє виконувати запити до пропрієтарних LLM, використовуючи єдиний інтерфейс, незалежно від конкретної моделі та компанії, що надає доступ до неї.

Для обраних open-source моделей було вирішено адаптувати публічний репозиторій для отримання натренованих LLM та датасетів для них – "Huggingface", а також бібліотеку, розроблену цим сервісом, "transformers".

Завдяки даному репозиторію, можливо вести індексування найпопулярніших LLM з часом, за допомогою "transformers" забезпечується можливість локального запуску більшості моделей безпосередньо з репозиторію завдяки спеціальному програмному інтерфейсу для їх використання.

На основі концепції проекту створено діаграму варіантів використання, результат наведено на рис. 1. Технічна сторона виконання досліджень

вибраних LLM моделей реалізована у вигляді клієнт-серверної веб-додатки зі спрощеним графічним інтерфейсом користувача.

Для впровадження основної функціональності розробленого проекту обрано мову програмування Python версій 3.7.12, що дозволяє використовувати зручні колекції даних та інтегрувати бібліотеки з обробки та аналізу текстових даних. Для впровадження ряду функціональних можливостей у рамках веб-програми необхідно створити інтерактивну взаємодію між користувачем та веб-сторінкою, з цією метою використана мова програмування JavaScript. Для побудови каркасу веб-застосування та покращення роботи з БД вирішено використовувати фреймворк Django.

Для зберігання даних прийнято рішення використовувати реляційну БД PostgreSQL, створена БД з 4 таблиць для зберігання метаданих про моделі, результати експериментів, набори гіперпараметрів і датасетів.

Для забезпечення більш простого управління залежностями та версіями, можливості запускати пропоновану платформу на багатьох платформах та логічного розподілу архітектури системи вирішено використовувати Docker та Kubernetes.

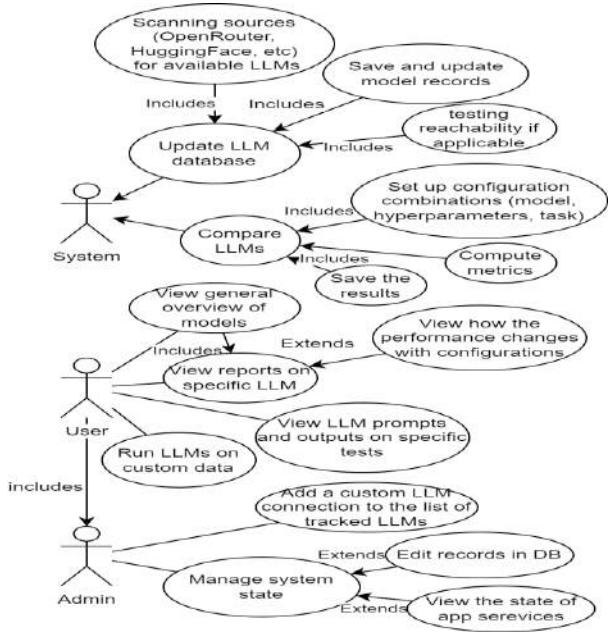

Рисунок 1. *Діаграма варіантів створеного проекту системи*

Особливістю програмної імплементації проекту є широкий перелік конфігурацій задачі аналізу текстових постів для виявлення ботів у SN,

інтеграція з платформою X для отримання доступу до даних відкритих постів та для автоматизації оцінки LLM, а також функціонал автоматичного трекінгу моделей у відкритих репозиторіях, що обумовлено раніше описаною зміною їх якості із різними версіями.

Веб-додаток використовується таким чином: користувач взаємодіє із системою через веб-додаток, він може вибирати різні роути, кожен з яких надає функціонал, необхідний для задоволення функцій користувача.

Процес експериментування проводиться автоматично, система вибирає комбінації LMM, конфігурацій і завдань, у яких слід провести експерименти, зберігаючи отримані результати таблиці БД. Адміністратор - користувач, який проводить хостинг розробленої платформи для проведення експериментів у ручному режимі, задаючи конфігурації, використовуючи та тестуючи новітні моделі та методи, вносячи модифікації в методології проведення експериментів, а також редагуючи відкритий код розробленого веб-додатку.

Діаграма компонентів системи, задіяних у процесі розгортання, наведена на рис.2. Кластер складається з наступних елементів (кожен вузол – окрема віртуальна чи реальна машина):

1. Вузол управління. На цьому вузлі виконуються завдання, пов'язані з оркестрацією завдань, передачею повідомлень та управління кластером. Саме він розміщує елемент управління кластером kubernetes (“Control Plane”), сервер брокера повідомлень Kafka для комунікації даними та повідомленнями між окремими аплікаціями в кластері, та сервер оркестрації робіт Apache Airflow для виконання завдань з трекінгу та оцінки мовних моделей. Оскільки цей вузол є найважливішим у кластері, вирішено не розміщувати на ньому тільки той код, який взято з довірених бібліотек (Kubernetes, Kafka, Apache Airflow).

2. Вузол постійних БД. На цьому сайті розміщуються сервери баз даних, необхідні для кластера. Розміщуючи їх на окремій машині, дозволить оптимізувати цей вузол під постійне збереження даних.

3. Вузол веб-сервера. На цьому вузлі знаходиться веб-сервер для взаємодії користувачів із системою, а також метод створення повідомлень Kafka (це необхідно для функціоналу, де користувач вимагає виконання генерації деякою моделлю).

4. Вузол трекінгу та оцінки мовних моделей. На цьому вузлі розміщуються контейнери та методи виконання трекінгів мовних моделей та їх оцінки. До того ж, ці методи запускаються виконавцем завдань Airflow, якщо оркестратор вирішує їх запустити, і вузол також розміщує метод створення повідомлень Kafka (це необхідно для функціонала, де потрібно запросити створення тексту).

5. Вузол запуску мовних моделей. На вузлі розміщуються методи генерації тексту мовними моделями, а також метод отримання повідомлень Kafka про генерацію тексту.

6. Інтернет. Кластеру необхідно мати доступ до Інтернету для обслуговування запитів користувачів, знаходження та трекінгу мовних моделей, а також посилання на API запитів на генерацію тексту для оцінки моделей.

7. Хост. Вузлу керуванням кластера необхідно мати доступ до хост-машини, з якої він отримуватиме команди з керування сервером.

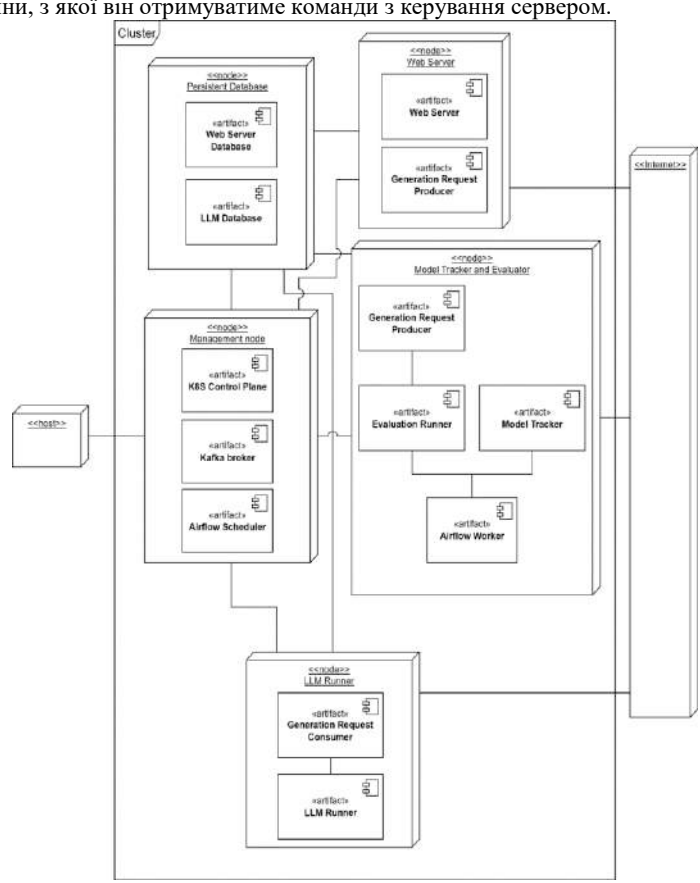

Рисунок 2. *Діаграма послідовності дій системи*

Для створення системи було вирішено в першу чергу реалізувати 4 головні сторінки, що відповідають функціональним вимогам: сторінка з навігацією, сторінка з переглядом метрик конкретної LLM на задачах аналізу тексту, сторінка з переглядом відповідей LLM щодо набору постів для задачі класифікації, сторінка з можливістю внесення користувачем даних щодо завдання та отримання результатів роботи LLM. Дані сторінки дозволять отримати ключові результати оцінки роботи моделей і відповідатимуть за більшу кількість інформації, яку можна відобразити у звітах. За цими вимогами було розроблено мокап інтерфейсу (рис.3, 4). Для збереження та управління даними вирішено розробити базу даних (БД) на базі використання NoSQL підходу, який на відміну від реляційної бази, е дозволить мати більшу

гнучкість стосовно структури БД під час розробки, що допоможе у випадку зміни вимог та структури проекту. У якості СУБД було обрано MongoDB через її велику популярність та наявність документального матеріалу.

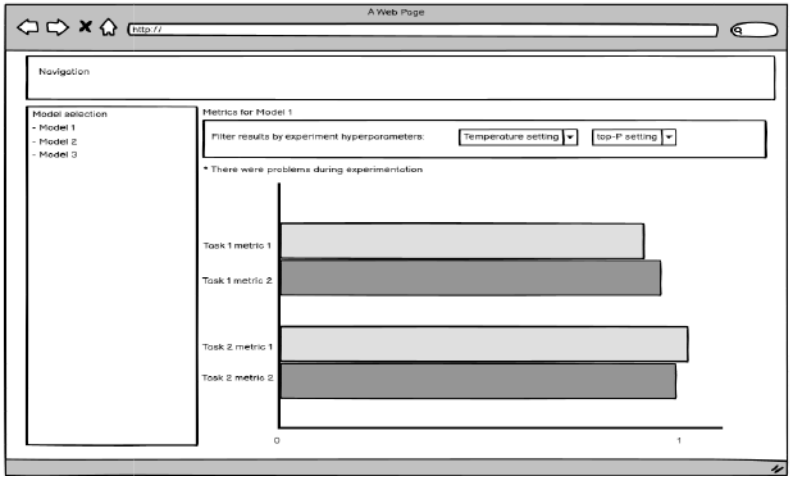

Рисунок 3. *Прототип інтерфейсу головної сторінки системи*

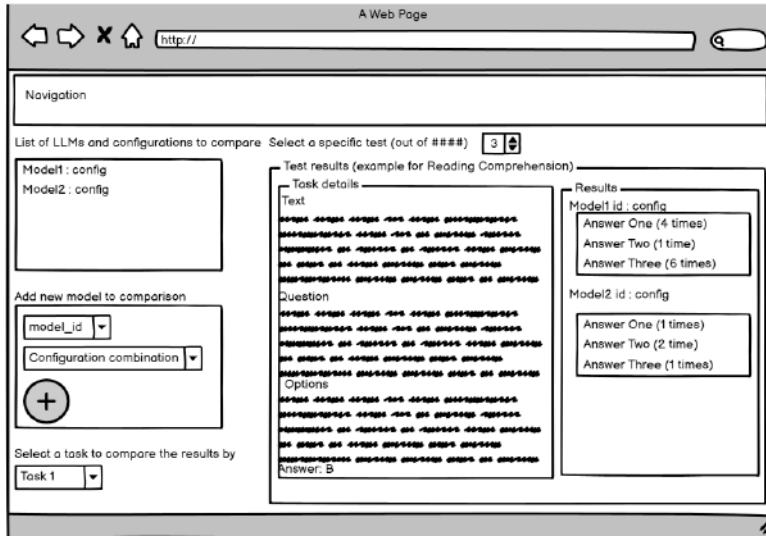

Рисунок 4. *Прототип інтерфейсу сторінки з переглядом результатів роботи LMM на тестах*

Структура бази даних для запропонованої платформи та їх евалуації зображена на діаграмі (рис.5).

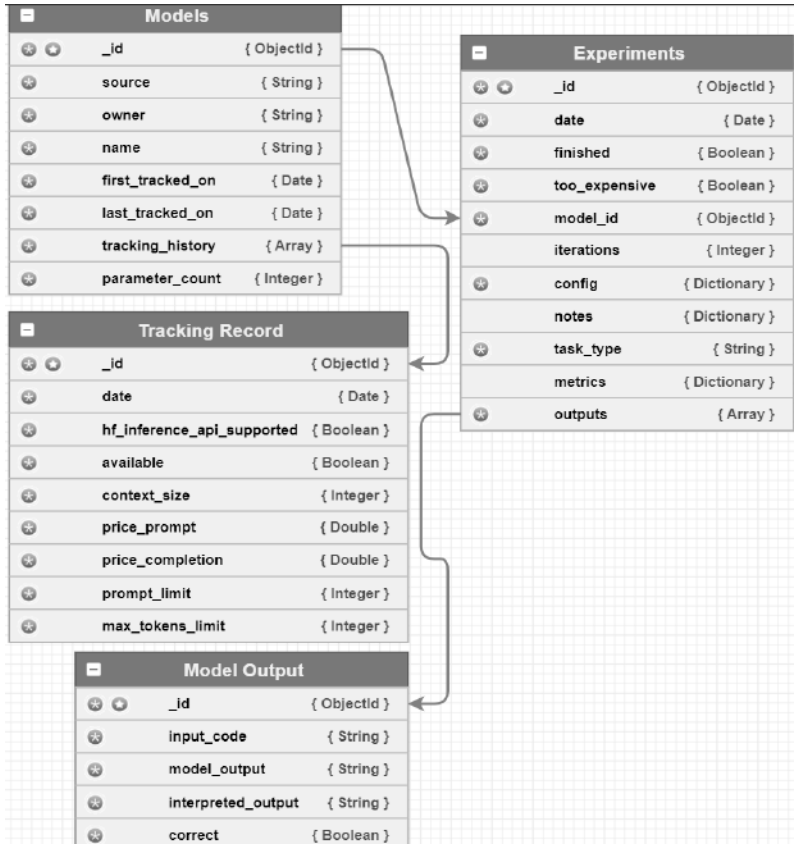

Рисунок 5. *Структура колекцій створеної БД*

Структура бази даних розподілена на дані частини, починаючи з колекції моделей, яка складається з наступних полей не зважаючи "_id":

1. Source - джерело з якого є доступ до язикової моделі. Цей параметр необхідний для визначення того, яким саме методом буде виконана генерація тексту..
2. Owner - компания чи користувач, який надає доступ до моделі
3. Name - назва моделі.

4. First_tracked_on - дата, коли система вперше здійснила перший трекінг моделі.
5. Last_tracked_on - дата, коли система вперше здійснила останній трекінг моделі.
6. Tracking_history - масив об'єктів з записами о трекінгу моделі.

Об'єкти записів трекінгу:

1. Date - дата створення запису трекінгу.
2. Hf_inference_api_supported - чи підтримує модель використання API Huggingface для генерації тексту (має сенс лише у випадку коли джерелом є huggingface).
3. Available - чи є доступ до моделі узагалі.
4. Context_size - максимальний розмір контексту який підтримує модель.

Запис експерименту:

1. Date - дата початку експерименту.
2. Finished - чиє експеримент завершеним.
3. Too_expensive - чи був експеримент завершений через перевищення ліміту на ціну генерації.
4. Model_id - ID язикової моделі, над якою був проведен даний експеримент.
5. Iterations - На даний момент не використовується. Може бути використаний для позначення кількості ітерацій над набором даних, у тому випадку де це релевантно.
6. Config - словник, визначаючий конфігурацію яка була використана для генерації тексту.
7. Notes - словник, не використаний на даний момент, але може бути використаний для позначення спеціальних заміток.
8. Task_type - позначає тип задачі над якою був проведений відповідний експеримент.
9. Metrics - словник з метриками, які оцінюють якість моделі в завершеному експерименті.
10. Outputs - масив з детальними результатами роботи усіх мовневих моделей.

Запис результату роботи моделі:

1. Input_code - код, який ідентифікує який саме запис з якого набору даних був використаний для генерації отриманого тексту.
2. Model_output - текст, який був згенерований моделлю.
3. Interpreted_output - те, як згенерований текст був інтерпретований при оцінки моделі (наприклад, згенерований текст "(A)" в задачі на вибір варіанту відповіді буде інтерпретовано як "A").
4. Correct - чи збігається інтерпретований результат з очікуваним в даному тесті, якщо це актуально.

Потрібно врахувати те, що в обраній NoSQL базі даних ("MongoDB") об'єкти записів трекінгу, експериментів та виводів моделей можуть бути як окремими колекціями, так і частинами іншого об'єкта (наприклад, записи трекінгу можуть бути частинами запису язикової моделі).

На даний момент, для гнучкості розробки було вирішено створити дві окремі колекції - моделей та експериментів, з вбудованими записами трекінгу та виводів моделей.

Проект структурований за каталогами так:

– experiment_running, що містить програмну логіку імплементації процесу запуску моделей LLM;

– frontendapp, містить реалізацію процесів завантаження та відображення даних в інтерфейсі користувача за різними роутами через RESTAPI запити. Містить підкаталоги migrations та templates для зберігання скриптів виконання міграції даних у БД та html сторінок розмітки уявлень;

– llpfrontend, містить програмну логіку роботи сервісних функцій фреймворку Django для базового функціоналу роутингу та спільної роботи адміністративної панелі;

– model_running, імплементує ключовий програмний функціонал забезпечення процесів побудови моделей LLM, завантаження наборів даних та їх розбиття на навчальну та тестову підвиборки, лематизацію та токенізацію, нормалізацію даних, формування окремих промптів, функції зворотного виклику для обробки та збереження метрик оцінки моделей LLM при десері побудові;

– model_tracking, реалізує функціонал відстеження змін у метриках формованих моделей;

– mongodb, містить конфігурацію побудови схеми БД для зберігання даних про LLM.

Деталізована структура файлів у пакетах проекту системи наведена на рис.6.

Окремо слід зазначити, що базова бізнес логіка роботи системи з управління потоками та рендерингом даних у рамках запуску веб-додатка знаходиться в каталозі frontendapp, у тому числі файли, що генеруються при ініціалізації проекту: admin.py, призначений для адміністративних функцій, зокрема для процедури реєстрації моделей, які використовуються в інтерфейсі адміністратора; apps.py, визначає базову конфігурацію програми та не стандартні параметри її запуску; models.py, зберігає визначення моделей, які описують дані, що використовуються в додатку; tests.py, описує модульні тести програми, моки, стаби та заглушки; views.py визначає функції, які отримують запити користувачів, обробляючи їх і повертаючи результат через сервер.

Запуск файлу здійснюється в командному рядку через менеджер пакетів pip. Перераховані залежності, що використовуються для шаблонизації (jinja2), інтеграції коду (Jupiter_core, ipython та ін.), підключення та взаємодії з БД (pymongo, sqlparse та ін.), обробки даних, об'єктів та результатів роботи ІНС (keras, tensorboard, torch, huggingface та ін), підтримки асинхронних запитів та автоматизації розгортання проекту (async-timeout, executing, pyyaml та ін), аналізу та виведення даних (numpy, pandas, matplotlib та ін), а також бібліотеки утилітарної спрямованості (кешування, асинхронність) та ін).

Рисунок 6. *Деталізована структура файлів у пакетах проекту системи*

В рамках проекту сформовано механізм автоматичної експериментальної оцінки LLM, та у даному розділі було описано одну ітерацію даного алгоритму. Його метою є оцінка LLM на деякій задачі, та збереження метрик у

базі даних. При розробці алгоритму, було визначено декілька факторів, які необхідно врахувати: 1.Необхідно проводити експериментування на кожній LLM періодично, бо, як було зазначено раніше, LLM можуть бути оновлені з часом. Це особливо стосується пропрієтарних LLM, про оновлення яких компанія-розробник може не об'являти. 2. Необхідно підготувати запити до LLM (комбінація інструкцій та тексту стосовно конкретної задачі) відповідно до особливостей LLM. Наприклад, деякі моделі оптимізовані до виконання інструкцій у запиті, якщо ці інструкції були виділені певним чином. Через це, при складанні запитів для експерименту, необхідно враховувати особливості певної моделі та відповідним чином складати запит.3. Необхідно розраховувати ціну. 4. Необхідно враховувати те, що різні LLM мають різні методи їх виконання. Тобто, мала open-source LLM може бути завантажена та використана локально, тоді як більша LLM може бути використана через спеціальний сервіс, а пропрієтарні LLM можуть бути використані лише через деякі спеціалізовані API. Через це, необхідно обирати метод доступу в залежності від LLM. 5. Необхідно розраховувати вартість виконання експерименту, та звіряти її з допустимою ціною.

4. Дослідження роботи системи

У таблиці 1 представлені метрики для завдання визначення ботів в SN відповідно до історії їх публікацій на базі TL адаптованих LLM GPT2, bloomz-1b1, mistral-7b і без нього (в останньому випадку результати були в 3-4 рази гірше в порівнянні з адаптованим варіантом). Були проведені експерименти з двома типами запиту ("prompt type") - тип запиту "with explanation" у якому є пояснення щодо того, які публікації зазвичай роблять боти (рекламні пости, повторювані пости, посилання на новини, або надто монотонні пости), та запит "without explanation" у якому немає даного пояснення. Розмір доповненого набору даних складає 6760000 записів. Аналізуючи отримані результати, можна зробити висновок, що різні LLM мають значно різну якість незалежно від їх розміру. Наприклад, за метрикою Recall видно, що bloomz-1b1 відзначає користувачів як бот частіше ніж інші, тому її не має сенсу використовувати на практиці, тоді як модель mistral-7b, яка має той же розмір, виявила значно більшу точність (більше 0.9) як класифікатор ботів. Також mistral-7b за метриками відповідає кращим результатам, ніж gpt2 і bloomz-1b1, при цьому майже в 7 разів більше bloomz-1b1. Аналізуючи метрики BMM, що найбільш якісно впоралися з даним завданням (mistral-7b і bloomz-1b1), зроблено висновок, що точне знаходження ботів в онлайн-мережах завдяки LMM можливо проводити, проте для цього потрібно тонке налаштування та передобробка даних для отримання більшого ступеня адекватності та узагальнюючої здатності моделі, при цьому чутливість використаних LLM до ряду гіперпараметрів у разі проведених експериментів невелика. Тобто. під час формування моделей попередньо слід перевірити дані узгодженість. На додаток наголосимо на тому, що наявність у запиті пояснення того, якими

зазвичай є публікації від аккаунтів-ботів, негативно вплинула на якість LLM усім метрикам. Тому зроблено висновок, що при інжинірингу запитів слід враховувати те, що новий змінений формат запиту може погіршити якість LLM, навіть якщо він має додану інформацію з наміром поліпшити якість моделі, тому необхідно проводити окрему процедуру оцінки нових форматів запиту. Ітерація показників візуалізації на моделі GPT2 наведена на рис.7.

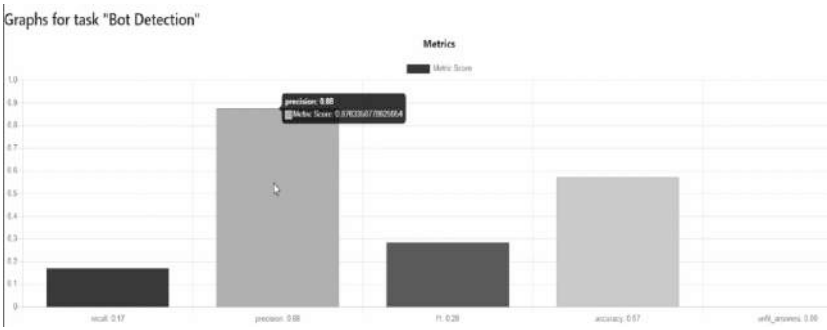

Рисунок 7. *Ітерація показників візуалізації на моделі GPT2*

Таблиця 4

Результати метрики LLM щодо завдання ідентифікації ботів у SN

LMM	Prompt Type	Recall	Precision	F1	Accu-racy	Unfit Answers
gpt2	Without Explanation	0,87	0.66	0.22	0.75	15
gpt2	With Explanation	0,84	0.64	0.26	0.73	10
bloomz-1b1	Without Explanation	0.79	0.79	0.31	0.7	25
bloomz-1b1	With Explanation	0.77	0.78	0.28	0.72	23
mistral-7b	Without Explanation	0.95	0.9	0.14	0.92	2
mistral-7b	With Explanation	0.93	0.87	0.12	0.91	3

На рис.8 наведено результат перегляду фрагмента постів одного з користувачів (записів датасета), коли модель видає різні класи в залежності від довжини та змістовності запиту.

User's post history (user ID = 104eee2839f69f59af377fec70eaaf7b):

- IT Security Analyst: IT Security Analyst – Berkshire – Permanent – SOC, Threat, vulnerability, SIEM, risk, malware Outsource UKs cyber team have an exciting opportunity for an IT Security Analyst to join an organisation that are currently insourcing... https://t.co/sKOc98PIad
- CRM Developer (Dynamics): I have an immediate requirement for a Dynamics CRM Developer to join my client on an initial 6 month contract (Outside of IR35) . You will be working on a major Dynamics 365 project for a leading organisation based in... https://t.co/NEwKWpVeI5 https://t.co/mTipJar3V4
- The election overseer for critical Palm Beach County says there is no way the recount for 3 races will be finished by Thursday's deadline: The election overseer for a critical county in Florida confirmed to CNN on Sunday what observers in both parties... https://t.co/aXmntXQhBN https://t.co/VoI9MzVkpQ
- Software Engineer – C++ / C# / UML: Software Engineer – C++ / C# / UML – Various Levels Commutable from Uxbridge / Slough / Watford / High Wycombe / Staines / Twickenham A market leading Global manufacturer is seeking to recruit a Software Engineer to... https://t.co/wA0S8fmqZU https://t.co/RmSKLPX8ME

Рисунок 8. *Приклад публікації бота, зібраний у набір даних*

Користувач, текст якого наведено вище, на самому місці є ботом, часто публікує пости - вакансії на роботу та пости на IT тематику. Можливо припустити, що даний користувач є роботом іншої компанії для рекрутингу нових співробітників і не має негативного ефекту в рамках SN, т.к. контролюється співробітниками організації. Модель GPT2 класифікувала даний користувач як бота з впевненістю на 77%, bloomz-1b1 з 88%, mistral-7b з 94%. У цьому прикладі можна наглядно дослідити правильність роботи моделі LLM за класифікацією тексту користувача як належного боту. Це можна пояснити тим, що завдання формалізації в запиті описує публікації постів від ботів як такі, які містять явну рекламу (у тому числі посилань), повторювані та близькі за контекстом фрази в поштах, заголовки новин у різних реєстрах з використанням високорелевантних анкорів або занадто звучать. монотонно, без визнань зміни тональності тексту. Слідкуйте також за тим, щоб більша частина публікацій відзначила пости від даного користувача звучати більш енергійно.

5. Висновки

У результаті проведених досліджень виконано застосування та адаптація існуючих більших мовних моделей для обробки та інтелектуального аналізу різноманітних текстових більших обсягів при виявленні ботів у соціальних мережах. Розроблене веб-прикладення дозволяє підключати, вибирати, відстежувати оновлення в API та адаптувати вибрані LLM для вхідних наборів даних, автоматизуючи всі етапи аналізу даних в окремих пайплайнах за допомогою AirFlow та інших технологій. Адаптовані на базі TL моделі GPT2, bloomz-1b1, mistral-7b в цілому успішно справляються із задачами виявлення ботів у SN за їх текстовим постам, найбільша точність досягається моделлю mistral-7b без функції видачи пояснень. Виходячи з отриманих результатів дослідження, можна зробити висновок про те, що наявність додаткових

пояснень при аналізі агрегованих різнорідних текстів користувачів, обсяг яких перевищує довжину окремого місця, часто накладає додаткові обмеження на роботу ANN, обмежуючи їх узагальнюючу здатність. У разі відсутності даного параметра робота моделі ANN є більш середньозв'язаною, однак цей ефект може бути пов'язаний зі специфікою використовуваного набору даних.

У подальшому раціональним шляхом дослідження є емпіричний пошук ступенів впливу даного параметра на робочу модель і виявлення його відносного значення в довільній формі з урахуванням підходу TL.

6. Література

[1] N. Rudnichenko, V. Vychuzhanin, N. Shibaeva, I. Petrov, T. Otradskya, Intelligent Data Clustering System for Searching Hidden Regularities in Financial Transactions, in: 11-th International Conference «Information Control Systems & Technologies» (ICST-2023) CEUR-WS, 3513, 2023, pp. 163-176.

[2] V. Vychuzhanin, N. Rudnichenko, Z. Sagova, M. Smieszek, V. V. Cherniavskyi, A. I. Golovan, M. V. Volodarets, Analysis and structuring diagnostic large volume data of technical condition of complex equipment in transport. IOP Conference Series: Materials Science and Engineering, Volume 776, 24th Slovak-Polish International Scientific Conference on Machine Modelling and Simulations - MMS 2019, 3-6 September 2019, Liptovský Ján, Slovakia, 2019 pp.1-11. DOI:10.1088/1757-899X/776/1/012049.

[3] N. Rudnichenko, V. Vychuzhanin, I. Petrov, D. Shibaev, Decision Support System for the Machine Learning Methods Selection in Big Data Mining, in: Proceedings of The Third International Workshop on Computer Modeling and Intelligent Systems (CMIS-2020), CEUR-WS, 2608, 2020, pp. 872-885.

[4] C. Segalina, D. Cheng, M. Cristani, Social profiling through image understanding: Personality inference using convolutional neural networks, Computer Vision and Image Understanding 156 (2017) 34–50.

[5] F. Liu, Zh. Li, Ch. Yang, D. Gong, H. Lu, F. Liu, SEGCN: a subgraph encoding based graph convolutional network model for social bot detection, Scientific Reports 14 (2024). DOI: 10.1038/s41598-024-54809-z.

[6] M. Zhou, W. Feng, Y. Zhu, D. Zhang, D. Yuxiao, J. Tang, Semi-Supervised Social Bot Detection with Initial Residual Relation Attention Networks, Machine Learning and Knowledge Discovery in Databases: Applied Data Science and Demo Track (2023) 207-224. DOI: 10.1007/978-3-031-43427-3_13.

[7] S. Gera, A. Sinha, T-Bot: AI-based social media bot detection model for trend-centric twitter network, Social Network Analysis and Mining 12 (2022). DOI: 10.1007/s13278-022-00897-6.

[8] Z. Ellaky, F. Benabbou, S. Ouahabi, N. Sael, A Survey of Spam Bots Detection in Online Social Networks, in: Conference: 2021 International Conference on Digital Age & Technological Advances for Sustainable Development (ICDATA), 2021, pp. 58-65. DOI: 10.1109/ICDATA52997.2021.00021.

[9] N. Sadeghi, N. Riahi, Comparison of the effect of the generative model on the performance of deep neural networks and transformer in contextual social bot detection (2023). DOI: 10.21203/rs.3.rs-2556040/v1.
[10] E. Kheir, R. Daouadi, R. Rebaï, I. Amous, Bot Detection on Online Social Networks Using Deep Forest. Artificial Intelligence Methods in Intelligent Algorithms (2019) 307-315. DOI: 10.1007/978-3-030-19810-7_30.
[11] S. Pulipati, Malicious Social Bots Detection in Online Social Networks with Using Ensemble Model 14 2510 (2022). DOI:10.9756/INT-JECSE/V14I2.232.
[12] P. Pham, L. Nguyen, B. Vo, U. Yun, Bot2Vec: A general approach of intra-community oriented representation learning for bot detection in different types of social networks. Information Systems 103 (2021). DOI:10.1016/j.is.2021.101771.
[13] M. Duddu, D. Mahesh, Detection of Social Bots in Twitter Network, in: Proceedings of International Joint Conference on Advances in Computational Intelligence, 2023, pp.655-668. DOI:10.1007/978-981-99-1435-7_53.
[14] M. Mendoza, E. Providel, M.L. Santos, S. Valenzuela, Detection and impact estimation of social bots in the Chilean Twitter network, Scientific reports 14 6525 (2024). DOI:10.1038/s41598-024-57227-3.
[15] N. Rudnichenko, V.Vychuzhanin, T. Otradskya, I. Petrov, I. Shpinareva, Hybrid Intelligent System for Recognizing Biometric Personal Data, in: Proceedings of the 3rd International Workshop on Computational & Information Technologies for Risk-Informed Systems (CITRisk 2022) the co-located with XXII International scientific and technical conference on Information Technologies in Education and Management (ITEM 2022), CEUR-WS, 3422, 2023. pp. 74-85.
[16] M. Zhou, D. Zhang, W. Dan, G. Yuandong, Y. Yangli-ao, T.J. Dong, LGB: Language Model and Graph Neural Network-Driven Social Bot Detection (2024). DOI: 10.48550/arXiv.2406.08762
[17] S. Ozdemir. Quick Start Guide to Large Language Models: Strategies and Best Practices for Using ChatGPT and Other LLMs (2023).
[18] A. Wang, SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems (2019). DOI:10.48550/arXiv.1905.00537
[19] J. Devlin, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Google AI Language 4 (2018) 29. DOI:10.48550/arXiv.1810.04805
[20] H. A. Naveed, Comprehensive Overview of Large Language Models (2023).
[21] M. Grohs, Large Language Models can accomplish Business Process Management Tasks (2023). DOI:10.48550/arXiv.2307.09923
[22] R.E. Turner, An Introduction to Transformers (2023). DOI:10.48550/arXiv.2304.10557
[23] Data set for bot user classification. URL: https://zenodo.org/records/3692340
[24] T. Wolf Transformers: State-of-the-Art Natural Language Processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020, pp.38-45.
[25] M. Sanghoon, H. In, J. Wonik, C.M. Jae, R. Jisu, S. K. Dae, K. Kee-Eung, J. Changwook, PAC-Net: A Model Pruning Approach to Inductive Transfer Learning, in: Proceedings of the 39th International Conference on Machine Learning, PMLR 162, 2022. DOI:10.48550/arXiv.2206.05703

AN INTELLIGENT SYSTEM FOR ANALYZING AND DETECTING TEXT BOT CONTENT OF GREAT IMPORTANCE IN SOCIAL NETWORKS

Ph.D. M. Rudnichenko ORCID: 0000-0002-7343-8076
Odesa Polytechnic National University, Ukraine
E-mail: nickolay.rud@gmail.com
Ph.D. N. Shibaeva ORCID: 0000-0002-7869-9953
Odesa Polytechnic National University, Ukraine
E-mail: nati.sh@gmail.com
Ph.D. T. Otradska ORCID: 0000-0002-5808-5647
Odesa College of Computer Technologies "SERVER", Ukraine
E-mail: tv_61@ukr.net
D. Shvedov ORCID: 0009-0002-4823-8782
Odesa Polytechnic National University, Ukraine
E-mail: frumle@ukr.net
Ph.D. I. Shpinareva ORCID: 0000-0001-9208-4923
Odesa Polytechnic National University, Ukraine
E-mail: iryna.shpinareva@onu.edu.ua
Dr.Sci. I. Petrov ORCID: 0000-0002-8740-6198
Odesa Polytechnic National University, Ukraine
E-mail: firmn@gmail.com

Abstract. *The paper addresses the challenges of analyzing and processing large volumes of heterogeneous natural language texts for the task of detecting bots on social networks using deep transfer learning methods, specifically large language models. It provides a detailed analysis of the specific characteristics and key aspects of structuring, processing, and analyzing text content, substantiates the relevance of the problem, and reviews existing approaches in the scientific literature. The paper highlights the advantages and potential applications of artificial neural networks and machine learning for automating the analysis of social network user posts. A description of the dataset selected for research is provided, along with a justification for the choice of artificial neural network language models, and an explanation of the use of transfer learning to adapt these models for bot detection. The technical tools and services employed to implement the functionality of the developed web application are described, and object-oriented models of the system are developed using UML, including use case and component diagrams. The software functionality, prototype pages, and graphical user interface are also outlined. The paper presents experimental results of the selected language models on an extended dataset, tested in modes both with and without text explanations. It analyzes the performance of adapted neural network models at a given stage, identifies the specifics of their operation, and suggests promising directions for further research and development to address the identified issues.*

Keywords: *Intelligent data analysis, text classification, social networks, text analysis, natural language processing, software development, bot detection, big data*