УДК 004.912:004.032.26
DOI https://doi.org/10.32689/maup.it.2025.1.20

Дмитро ПАВЛЮК
аспірант, Дніпровський національний університет імені Олеся Гончара, pavliuk_d@365.dnu.edu.ua
ORCID: *0009-0003-6670-6022*

Олег БАЙБУЗ
доктор технічних наук, професор, завідувач кафедри інженерії програмного забезпечення та інформаційних технологій,
Дніпровський національний університет імені Олеся Гончара, baibuz_o@365.dnu.edu.ua
ORCID: *0000-0001-7489-6952*

ДОСЛІДЖЕННЯ МОЖЛИВОСТЕЙ АВТОМАТИЗОВАНОГО АНАЛІЗУ ТОНАЛЬНОСТІ ТЕКСТІВ ЗА ДОПОМОГОЮ СУЧАСНИХ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ

Анотація. Мета дослідження. *У статті досліджено можливості автоматизованого аналізу політичних коментарів з використанням сучасних великих мовних моделей (LLM). Метою є розробка програмного рішення, яке класифікує текстові коментарі на два рівні: за емоційним забарвленням (позитивне, негативне, нейтральне) та за цільовим об'єктом реакції (подія, автор, стиль публікації, аудиторія). Оцінено ефективність застосування LLM для аналізу тональності політичних коментарів на основі даних з Telegram-каналів.*

Методологія. *Для досягнення мети було розроблено програмний прототип, який здійснює автоматичний аналіз текстів. Прототип використовує два виміри класифікації: емоційний тон і цільовий об'єкт реакції, з урахуванням специфіки політичного контексту. Вхідні дані представляють собою текстові публікації з Telegram-каналів та відповідні користувацькі коментарі, а результати класифікація відбувається за допомогою LLM з використанням підходу few-shot learning.*

Наукова новизна. *Розроблений прототип дозволяє здійснювати багатовимірну класифікацію текстів, що є нестандартним підходом у дослідженні політичного дискурсу, де важливо не лише визначити загальну тональність коментаря, а й з'ясувати, до кого чи до чого направлена реакція. Дослідження також пропонує стратегії покращення результатів класифікації, включаючи інтеграцію динамічних інструкцій та локалізацію навчання на україномовних даних, що може бути важливим кроком для підвищення ефективності використання LLM для політичного контенту в Україні.*

Висновки. *Результати дослідження показали, що LLM мають значний потенціал для виконання багатовимірної класифікації політичних коментарів, однак виявлено і обмеження, зокрема у виявленні сарказму та іронії, а також у роботі з локальними специфічними контекстами. Запропоновані стратегії покращення, такі як адаптація моделі до україномовних даних та використання динамічних підказок, дозволяють підвищити точність результатів. Дослідження підкреслює необхідність адаптації LLM до політичного контексту, зокрема для модерації контенту та соціологічних досліджень. У подальших дослідженнях необхідно зібрати більші та більш збалансовані датасети для більш релевантних та узагальнених результатів роботи розробленого програмного забезпечення.*

Ключові слова: *Класифікація тексту, великі мовні моделі (LLM), аналіз тональності, prompt engineering, Telegram API, user bot, автоматизація модерації, веб-розробка, few-shot learning.*

Dmytro PAVLIUK, Oleh BAIBUZ. EXPLORING THE CAPABILITIES OF AUTOMATED TEXT SENTIMENT ANALYSIS USING MODERN LARGE LANGUAGE MODELS

Abstract. The objective of the study. *This article explores the possibilities of automated analysis of political comments using modern large language models (LLM). The aim is to develop a software solution that classifies textual comments into two levels: by emotional tone (positive, negative, neutral) and by the target object of the reaction (event, author, publication style, community). The effectiveness of using LLM for sentiment analysis of political comments based on data from Telegram channels is evaluated.*

Methodology. *To achieve the goal, a software prototype was developed that performs automatic text analysis. The prototype uses two classification dimensions: emotional tone and target object of reaction, taking into account the specifics of the political context. The input data consists of textual posts from Telegram channels and corresponding user comments, and the classification results are achieved using LLM with a few-shot learning approach.*

Scientific novelty. *The developed prototype allows for multidimensional classification of texts, which is an uncommon approach in the study of political discourse, where it is important not only to determine the overall tone of the comment but also to identify who or what the reaction is directed towards. The research also offers strategies to improve classification results, including the integration of dynamic instructions and localization of training on Ukrainian-language data, which could be an important step in enhancing the effectiveness of using LLM for political content in Ukraine.*

Conclusions. *The results of the research showed that LLMs have significant potential for performing multidimensional classification of political comments. However, limitations were identified, particularly in detecting sarcasm and irony, as well as in working with local specific contexts. Proposed improvement strategies, such as adapting the model to Ukrainian-language data and using dynamic prompts, allow for improved accuracy of results. The research highlights the need to adapt LLMs to the political context, especially for content moderation and sociological research. Future research should focus on collecting larger and more balanced datasets for more relevant and generalized results in the operation of the developed software.*

Key words: *Text classification, large language models (LLM), sentiment analysis, prompt engineering, Telegram API, user bot, automation of moderation, web development, few-shot learning.*

Постановка проблеми. Сучасні інструменти аналізу тональності пропонують різні способи класифікації текстів, включаючи визначення емоційного забарвлення, виявлення токсичності або навіть ідентифікацію конкретних тем. Однак у контексті політичних дискусій у Telegram часто виникає потреба в більш деталізованому розумінні коментарів, яке виходить за межі загального позитиву/негативу. Наприклад, навіть якщо система точно визначає, що коментар має «негативний» тон, вона рідко вказує, що саме викликало таку реакцію: подія, автор публікації, стиль викладу чи інші фактори.

Для певних ситуацій, таких як аналіз політичної комунікації або модерація контенту, може бути корисним додатковий рівень деталізації. Без розрізнення нюансів аналітики, модератори або дослідники можуть втрачати важливі деталі. Наприклад, неможливість автоматично відокремити критику події від критики автора ускладнює:

– Роботу PR-фахівців, які мають реагувати на конкретні проблеми,
– Моніторинг суспільної довіри до окремих інституцій або осіб,
– Вивчення мовних патернів, пов'язаних із пропагандою чи маніпуляціями.

Тому актуальним залишається питання, чи можна доповнити існуючі підходи класифікації, додавши можливість визначати цільові об'єкти реакцій (подія, автор, публікація, аудиторія). Це дозволило б отримувати більш структуровані дані без радикальної зміни поточних рішень, а лише розширюючи їхні можливості для специфічних задач.

Мета дослідження. Розробити та протестувати прототип програмного рішення для аналізу політичних коментарів, яке автоматично завантажує тексти, класифікує їх за допомогою великих мовних моделей (LLM) на два рівні: емоційне забарвлення (позитивне, негативне, нейтральне) та цільовий об'єкт реакції (подія, автор публікації, стиль матеріалу, аудиторія), а також оцінити ефективність LLM у цьому контексті. Додаток має перевірити, чи здатні сучасні моделі точно ідентифікувати складні взаємодії в тексті, такі як одночасне схвалення однієї сутності та критику іншої, враховуючи політичну специфіку, культурні відсилки та ризики помилкового тлумачення (сарказм, іронія). Важливо визначити межі застосування LLM для багатовимірної класифікації та які методи (наприклад, динамічні підказки, few-shot приклади) найкраще підходять для політичного контенту українською/російською мовами. Результати дозволять зрозуміти, чи можна масштабувати подібні рішення для аналізу великих масивів даних без втрати якості, і чи економічно доцільно це з точки зору витрат на обчислювальні ресурси.

Аналіз останніх досліджень і публікацій. Останні роки відзначені активним дослідженням можливостей великих мовних моделей (LLM) у сфері аналізу тональності, зокрема в політичному контексті. Наприклад, робота «How to Train Your Stochastic Parrot: Large Language Models for Political Text» (2024) демонструє, що GPT-4 у режимі few-shot досягає к-узгодженості ≈0.72 з експертами-людьми при класифікації саркастичних твітів про рішення Верховного суду США[10]. Це свідчить про здатність LLM враховувати контекст і політичні реалії, однак автори зазначають, що точність різко падає без явного вказівки фону події в підказці.

Дослідження «Evaluating Large Language Models Against Human Annotators in Latent Content Analysis: Sentiment, Political Leaning, Emotional Intensity, and Sarcasm» (2025) порівнює 7 сучасних LLM (включаючи GPT-4, Gemini, LLaMA-2) з групою з 33 анотаторів у задачах визначення тональності, політичних уподобань та сарказму [3]. Результати показують, що моделі перевершують людей у стабільності оцінок (наприклад, однакові відповіді на однакові запити через час), але відстають у розпізнаванні іронії, особливо в кроскультурному контексті. Це підкреслює важливість локалізації підходів для конкретних мовних середовищ, як-от український сегмент Telegram.

У контексті політичного дискурсу, робота «Sentiment analysis using unsupervised learning for local government elections in South Africa» досліджує можливість аналізу сентименту та класифікацію користувачів як справжніх або ботів. Дослідження висвітлює практичні аспекти та обмеження при аналізі тональності твітів у локальному контексті (реакції на вибори у Південній Африці у Twitter/X) та обмеження та потенційні виклики при використанні unsupervised learning (наприклад, потенційні труднощі зі збалансованістю класів датасета та його актуальність) [6].

Важливий внесок у вивчення сарказму робить робота Gole et al. (2023), де fine-tuning GPT-3 на датасеті SARC 2.0 підвищив точність виявлення іронії до 81% (порівняно з 70% у zero-shot GPT-4) [5]. Автори наголошують, що навіть невеликі спеціалізовані датасети значно покращують якість класифікації, що актуально для аналізу політичних коментарів зі специфічними мемами чи референсами.

Викладення основного матеріалу дослідження. *Джерело отримання даних.* Перед тим як приступити до роботи необхідно визначити найбільш релевантну онлайн-платформу звідки будуть братися данні у контексті політичного дискурсу. Основні критеріями для вибору платформи були: охоплення аудиторії, характер публікацій та технічна можливість автоматизованного отримання даних. За результатами опитування Київського міжнародного інституту соціології (вересень 2023 року) найбільша кількість

респондентів (44%) отримують інформацію з новинних Telegram-каналів [1]. Крім того, значна кількість публікацій у текстовому вигляді та наявність інструментів для автоматизованого завантаження даних (TDlib, WTelegramClient та інші) робить платформу релевантною у контексті дослідження [2, 11, 12].

Архітектура та розробка програмного забезпечення. Програмне забезпечення було розроблено з використанням мови програмування C# та веб-фреймворку ASP.NET Core. Розроблений додаток складається з трьох сервісів:

– TelegramService: відповідає за отримання необхідних даних з Telegram;
– GptService: аналіз отриманих даних;
– TelegramAnalysisService: звертається до двох інших сервісів та агрегує отримані дані.

Логіка отримання даних. Важливо окреслити, що данні отримуються за допомогою User Bot акаунту Telegram: додаток надає можливість використовувати користувацький обліковий запис для доступу до Telegram API. При розробці TelegramService було використано nuget-пакет WTelegramClient. Сервіс включає у себе три основні компоненти:

– Спільна бібліотека TelegramSessionManagement;
– Web API додаток TelegramService;
– Консольний додаток TelegramHelper;

Спільна бібліотка TelegramSessionManagement включає логіку роботи з сесією WTelegramClient. Доступні наступні методи для ініціалізації клієнта User Bot:

CreateUserBotClientAsync(UserBotClientConfig clientConfig):

Створює новий екземпляр Client на основі переданої конфігурації, забезпечує існування папки для сесій та зберігає дані користувацької сесії та файлу конфігурації у файлову систему.

Параметр clientConfig типу UserBotClientConfig представляє собою об'єкт що зберігає дані необхідні для ініціалізації User Bot клієнта:

– ApiId: ідентифікатор API, який використовується для автентифікації.
– ApiHash: хеш API, який також використовується для автентифікації.
– PhoneNumber: номер телефону, прив’язаний до облікового запису Telegram.
– SessionPath: шлях до файлу, який використовується для збереження стану сесії.

CreateUserBotClientAsync()

Повертає існуючий екземпляр Client або завантажує його з файлу, якщо він ще не створений. При використанні цього очікується, що конфігурація впопередньо вже була збережена у файл.

Консольний додаток TelegramHelper використовується для авторизації User Bot клієнта, що використовує TelegramSessionManagement та WTelegramClient.

Додаток визначає такі консольні команди:

Команда login

Ця команда використовується для ініціалізації сесії у додатку Telegram. Також може бути використана для підключення до існуючої сесії, якщо вона вже була ініційована. Вона має наступні параметри:

– -i, --api_id (обов'язковий): ідентифікатор (api_id) Telegram API;
– -h, --api_hash (обов'язковий): хеш (api_hash) Telegram API;
– -n, --phone_number (обов'язковий): номер телефону Telegram API.

Команда connect

Ця команда використовується для підключення до поточної сесії Userbot. Вона не має параметрів.

Команда logout

Ця команда використовується для виходу з сесії Telegram.

Вона має наступні параметри:

– -i, --api_id (обов'язковий): ідентифікатор (api_id) Telegram API;
– -h, --api_hash (обов'язковий): хеш (api_hash) Telegram API;
– -n, --phone_number (обов'язковий): номер телефону Telegram API.

Веб-сервіс TelegramService представляє Web API для отримання даних з Telegram. Він містить один HTTP метод: GetPostWithReplies. Цей метод обробляє HTTP GET запити за маршрутом *api/telegram/postWithReplies*. Він приймає певний набір параметрів і повертає об'єкт PostWithReplies.

Параметри HTTP-запиту:

– ChannelName (обов'язковий): назва каналу в Telegram, з якого потрібно отримати публікацію;
– PublicationId (обов'язковий): ідентифікатор публікації, яку потрібно отримати;
– LoadRepliesOnComments (необов'язковий; за замовчування має значення false/0): прапорець, що вказує, чи потрібно завантажувати відповіді на коментарі інших користувачів. У контексті дослідження використовується значення false/0. Прапорець є важливим елементом препроцесингу вхідних даний, що дозволяє відфільтрувати коментарі у відповідь на інші коментарі як нерелевантні дані.

Після отримання відповіді у вигляді об'єкта *PostWithReplies*, метод повертає HTTP статус 200 (OK) разом з цим об'єктом. Якщо запит не авторизований, повертається статус 401 (Unauthorized).

Формат отримуваних даних (*PostWithReplies*):

– ChannelId (обов'язковий): ідентифікатор каналу в Telegram, з якого отримана публікація;
– PostId (обов'язковий): ідентифікатор публікації;
– PostText: текст публікації (необов'язковий);
– UsersReplies: список відповідей користувачів на публікацію (необов'язковий).

UserReplies представляє відповіді користувача і містить наступні властивості:

– UserId (обов'язковий): ідентифікатор користувача, який залишив відповіді;
– Replies (обов'язковий): список відповідей користувача.

Reply представляє окрему відповідь і містить наступні властивості:

– MessageId (обов'язковий): ідентифікатор повідомлення;
– Message (обов'язковий): текст повідомлення.

Сервіс аналізу тональності. GptService – це Web API сервіс відповідальний за комунікацію з Open AI Web API, визначення формату результатів аналізу та інструкції до LLM. Використовується метод POST за маршрутом */api/gpt/analysisBatch*. Цей метод виконує аналіз відгуків користувачів на публікацію. Він приймає запит, що містить текст публікації, відповіді користувачів, а також список операцій для аналізу. У відповідь повертає об'єкт, який містить результати аналізу для кожного користувача або відповідь зі 401 (не авторизовано) статус-кодом. Параметр: PostFeedbackAnalysisBatchRequest (тіло запиту): об'єкт, який містить дані для аналізу. Метод повертає PublicationSentimentReport: об'єкт, який містить результати аналізу.

Об'єкт PostFeedbackAnalysisBatchRequest предсталяє структуру запиту, який передається в метод POST */api/gpt/analysisBatch*. Його вміст:

– PostWithReplies (обов'язковий): містить текст публікації, відповіді користувачів, ідентифікатор каналу та ідентифікатор публікації;
– AnalysisOperations (обов'язковий): список операцій для виконання аналізу.

Складові PostWithReplies:

– string? PostText: текст публікації;
– List<UserReplies> UsersReplies: список відповідей користувачів (важливо при цьому уточнити той факт, що аналізуються виключно відповіді на публікацію, а не на повідомлення інших користувачів);
– string ChannelId: Ідентифікатор каналу;
– long PostId: Ідентифікатор публікації.

Можливі значення AnalysisOperations:

– GetAuthorAttitude: аналіз ставлення до автора;
– GetCommunityAttitude: аналіз ставлення до реакції спільноти;
– GetPostAttitude: аналіз ставлення до публікації;
– GetEventReaction: аналіз реакції на подію;
– GetWasPostUnderstood: аналіз того, чи була публікація зрозуміла;
– GetEnhancedPublication: отримання покращеної версії публікації.

Клас PublicationSentimentReport представляє структуру відповіді, яка повертається методом POST /api/gpt/analysisBatch. Його властивості:

– string? PostText: текст публікації, який аналізувався;
– List<UsersRepliesReport> UsersRepliesReports: список звітів про аналіз відповідей користувачів;
– string ChannelId (обов'язковий): ідентифікатор каналу, до якого належить публікація;
– long PostId (обов'язковий): ідентифікатор публікації;
– PublicationContext (обов'язковий): контекст публікації.

Структура UsersRepliesReport:

– UserReplies UserReplies (обов'язковий): відповіді конкретного користувача;
– long UserId: ідентифікатор користувача;
– List<Reply> Replies: список відповідей користувача;
– ReplySentimentReport? SentimentReport: результати аналізу для відповідей користувача;
– Attitude? AuthorAttitude: ставлення до автора;
– Attitude? CommunityAttitude: ставлення до реакції спільноти;
– Attitude? PublicationAttitude: ставлення до публікації;
– Attitude? EventAttitude: ставлення до події;
– string? Reasoning: обґрунтування аналізу.

У свою чергу Сутність Attitude включає у себе поле Sentiment яке зберігає такі значення:
– Positive,
– Negative,
– Neutral,
– Unknown

Структура PublicationContext:
– string? Context: додаткова інформація про контекст публікації;
– bool WasPublicationUnderstood: чи була публікація зрозуміла.

Бізнес-логіка відповідає за аналіз настроїв (sentiment analysis). Він працює з текстами публікацій, відповідями користувачів і виконує аналіз за допомогою OpenAI API. Основна мета сервісу – обробити текстові дані, виконати запит до великої мовної моделі та повернути структуровані результати. Клас відповідальний аналіз реакції включає два основних методи:

– Метод PerformSentimentAnalysis приймає текст публікації, список коментарів (очікуються коментарі від одного користувача) і список операцій для аналізу. Він формує запит до OpenAI API, отримує відповідь у вигляді JSON і перетворює її в об'єкт ReplySentimentReport. Результат містить інформацію про ставлення до автора, спільноти, публікації, події тощо.

– Метод PerformSentimentAnalysisBatch виконує аналіз настроїв для публікації та всіх відповідей користувачів. Він створює об'єкт PublicationSentimentReport, який містить результати аналізу для кожного користувача, а також контекст публікації. Для кожного користувача викликається окремий запит через згаданий метод PerformSentimientAnalysis. При цьому кожен запит до OpenAI Web API виконується паралельно, задля більш швидкого виконання та запобігання ризику перевищення обсягу контекстного вікна моделі (кожна модель має обмежену кількість вхідних та вихідних токенів на запит).

Відзначимо, що, задля зменшення кількості запитів до моделі та економії токенів, для кожного користувача повертається JSON-об'єкт типу ReplySentimentReport. Для вирішення проблеми ризику повернення некоректної JSON-схеми або неіснуючих значень поля Sentiment (наприклад, "none" замість "unknown") було використано Structured Outputs. OpenAI Structured Outputs – це функція API, яка забезпечує точну відповідність виводу моделі заданій JSON-схемі. Вона дозволяє отримувати структуровані відповіді, що відповідають заданому формату, без необхідності додаткової обробки [9].

У якості моделі для аналізу даних було використано модель GPT o4-mini[8]. Вибір зумовлений в першу чергу поєднанням просунутого обгрунтування та прийнятної вартості для практичного масового застосування [7].

Задля збільшення точності отриманих результатів було використано few-shot підхід (передача кількох прикладів класифікації через інструкцію). На даному етапі не розглядалося використання fine-tuning через потенційно надто широкий контекст дискурсу.

Інструкції до LLM. Для кожного користувача с його набором коментарів формується 3 послідовні інструкції:

1. Системна інструкція, яка загально окреслює спектр задач LLM та визначає логіку побудови вихідного об’єкту (звіту по аналізу тональності);

2. Інструкція з вхідними даними, яка включає текст публікації та коментарі користувача;

3. Набір задач для LLM, який вказує що саме необхідно класифікувати (ставлення до автора, події, публікації, спільноти). Будується динамічно, відштовхуючись від запиту дослідника (перелік значень у полі AnalysisOperations).

Системна інструкція: Your task is a texts classification.

I will provide you a text of the publication and the user's replies on it.

Also, I will provide classification tasks. You have a limited set of available answers: {категорії}. Create an object "report" of the type defined by JSON Schema (Structured Output). Fill out all requested fields. If you can’t determine a field, set it to "unknown". For all fields not mentioned – set value to null. Also, fill "report. reasoning" – write a brief reasoning (1-2 sentences per field filled).

Вхідні дані: The publication is the following: {текст публікації}. The user's replies on the publication are the following: {відповіді користувача}.

Приклад задачі класифікації (Ставлення до автора/каналу як автора): How can you classify user's opinion on the author or channel as an author? Is he/it a good publisher or not in user's opinion? You can rely on whether there are any accusations or direct insults against the author/channel.

Негативні приклади: «Один з найгірших новинних каналів»; «Скільки тобі заплатили за цю публікацію?»; «Ти, як завжди, публікуєш сміття»; «адмін клоун»; «провокаційне питання автору».

Позитивні приклади: «база»; «Автор базує»; «респект»; «харош».

Нейтральні приклади: «Це просто жах!»; «Круто»; «Ось воно як!».

«Насправді, все не так просто: …».

Assign the answer to the {назва поля} field.

Дані для аналізу. Для аналізу було виконано аналіз таких публічно доступних публікацій у Telegram:

– Міжнародний жіночий день: https://t.me/novinach/53860. Включає велику кількість дискусійних коментарів, які як підтримують свято, так ставлять під сумнів його доцільність.

– Зниження попиту на квіти на 8 березня: https://t.me/novinach/53861. Публікація подібна до першої, але є більш складною для аналізу: є ризик утотожнення LLM зниження попиту на квіти та 8 березня як таким, хоча підтримка зниження попиту на квіти може бути антагоністичною як і навпаки.

Важливо підкреслити наявність різко емоційно забарвленої лексики у багатьох коментарях – це є цінними даними для дослідження. У той же час дослідження не дає оцінку тій чи іншій позиції.

Результати дослідження. *Міжнародний жіночий день.* Додаток продемонстрував кращу точність у прогнозуванні ставлення читачів до автора публікації, спільноти та публікації у порівнянні з класифікацією вказаної події. Вірогідно, це пов'язано з відсутністю необхідності знання додаткового контексту для повного розуміння. У той же час, модель допускала помилки з розпізнаванням сарказму\іронії, особливо у специфічному контексті. Приклад:

Повідомлення: Скориставшись порадою Новинача, вступив у Фрайкор

Ставлення до автора: unknown

Ставлення до спільноти: unknown

Ставлення до публікації: positive

Ставлення до події: unknown

Даний коментар демонструє негативне ставлення як до публікації, так і до події (загони Фрайкор брали участь у вбивстві Рози Люксембург – людині тісно пов'язаної з Міжнародним жіночим днем [4]). Також, у даному випадку можна помітити залежність трактування ставлення до публікації від ставлення до події у даному випадку некоректного).

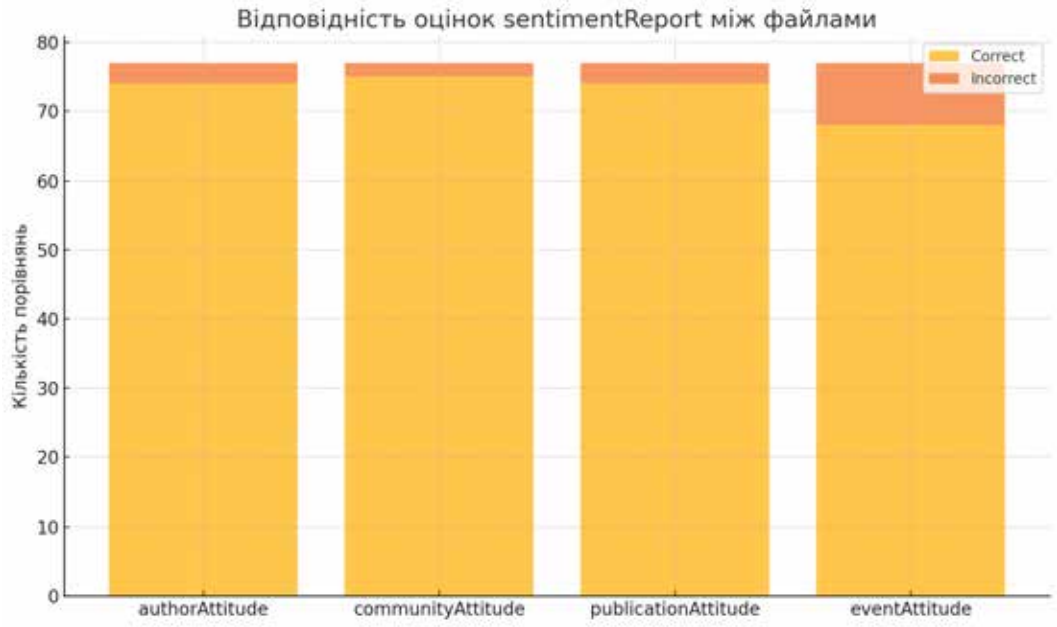

Рис. 1. Результати класифікації коментарів про Міжнародний жіночий день

На Рисунку 1 можемо бачити такі результати:

Ставлення до автора:

– Правильних: 74

– Неправильних: 3

Ставлення до спільноти:

– Правильних: 75

– Неправильних: 2

Ставлення до публікації:

– Правильних: 74

– Неправильних: 3
Ставлення до події:
– Правильних: 68
– Неправильних: 9

Зниження попиту на квіти. У порівнянні з публікацією про Міжнародний жіночий день точність класифікації ставлення до події зменшилася. Приклад:

Повідомлення: А дарма, бо свято має не комуністичне підгрунтя, а боротьбу жінок за права й рівності, коли вони намагалися вибороти право голосувати та гідні умови праці. Не бачу причин не просвяткувати й не привітати суспільно-активних жінок.

Ставлення до автора: unknown
Ставлення до спільноти: unknown
Ставлення до публікації: neutral
Ставлення до події: positive

У публікації йде мова про зниження попиту на квіти, а автор повідомлення демонструє невдоволення цією новиною. Вірогідно, LLM вважає підтримку Міжнародного жіночого дня підтримкою йому присвяченій публікації.

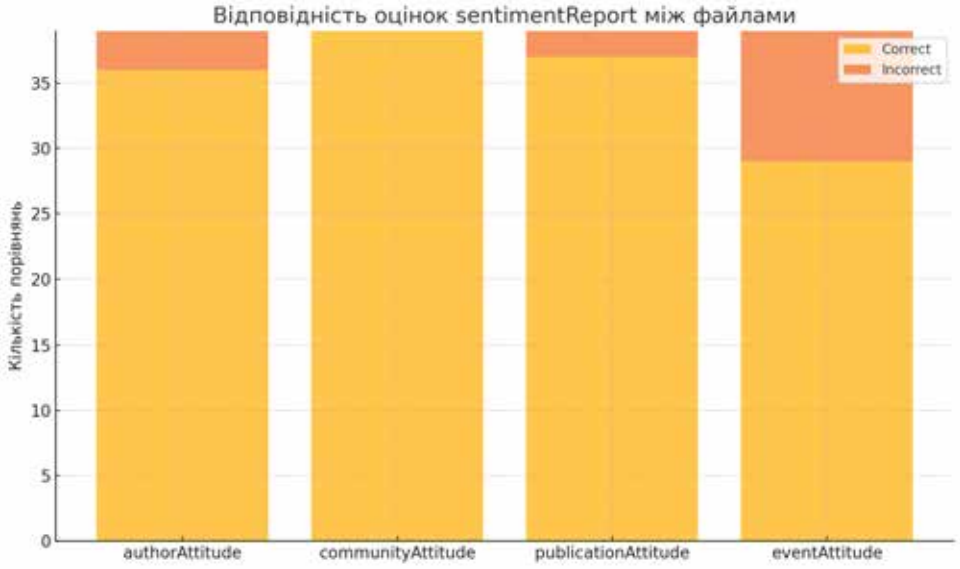

Рис. 2. Результати класифікації коментарів про зниження попиту на квіти

На Рисунку 2 можемо бачити такі результати:
Ставлення до автора:
– Правильних: 36
– Неправильних: 3
Ставлення до спільноти:
– Правильних: 39
– Неправильних: 0
Ставлення до публікації::
– Правильних: 37
– Неправильних: 2
Ставлення до події:
– Правильних: 29
– Неправильних: 10

Важливо зазначити, що у більшості випадків користувачі першочергово реагують на подію та публікацію, а реакція на інформаційний ресурс та на спільноту може бути класифікована як нейтральна або невідома.

Висновки. LLM демонструють потенціал для багатовимірної класифікації, але потребують явного вказівки цільових об'єктів (подія, автор тощо) у підказках задля запобігання підміни різних сутностей, зокрема згаданих подій та задіяних, але не тотожних понять.

Few-shot підходу часто з загальними прикладами часто може бути недостатньо: задля запобігання використання великих датасетів має сенс використовувати локально-специфічні зразки для few-shot класифікації.

Fine-tuning на вузьких датасетах (наприклад, україномовних політичних коментарях) може покращити точність, але потребує додаткових ресурсів. Тема політичного дискурсу досить розлога, тому збір датасетів може бути досить витратним завданням.

Тим не менш, у подальшому дослідженні необхідна робота з більшою кількість даних та робота над формуванням повноцінного збалансованого датасету. У задіяних даних найчастіше зустрічалися реакції на публікацію та подію, і у той же час реакція на автора публікації\медіа була найчастіше нейтральною\невідомою (хоча, важливо зазначити, що точність класифікації була вищою ніж для публікації та події у зв'язку меншою необхідність розуміти локальний контекст).

Вірогідно, необхідно також розробити більш чіткий та гнучкий підхід для розрізнення класів «neutral» та «unknown». Модель не завжди схильна до «визнання» того що вона може не знати повний контекст публікації або коментаря до неї. Інакше кажучи, не завжди зрозуміло чи потрібна додаткова підказка, тому що модель може показувати власне бачення неочевидних або вузько спеціалізованих знань.

Недостача специфічних локальних знань LLM вимагає додаткових механізмів корекції (наприклад, балансування підказок або використання динамічних підказок в залежності від контексту).

Таким чином, сучасні LLM підходять для вирішення задач складної класифікації тексту, але необхідно враховувати локальний контекст дослідження та розуміти обмеження моделей. Крім того, для подальшого дослідження необхідна більша кількість даних для більш репрезентативних та узагальнених результатів.

Список використаних джерел:

1. Київський міжнародний інститут соціології. Results of the all-Ukrainian survey for the European Union Advisory Mission in Ukraine. 2023. URL: https://kiis.com.ua/?lang=ukr&cat=reports&id=1307&page=1 (Дата звернення: 8 листопада 2024).

2. Павлюк Д. І., Байбуз О. Г. Ресурси збору даних для навчання з учителем для прогнозування суспільних настроїв. У: Кісельова О. М., ред. Математичне та програмне забезпечення інтелектуальних систем (МПЗІС-2024): Тези доповідей XXII Міжнародної науково-практичної конференції. Дніпровський національний університет імені Олеся Гончара. 2024. URL: https://ir.lib.vntu.edu.ua/bitstream/handle/123456789/43697/167315.pdf

3. Bojic L., Zagovora O., Zelenkauskaite A., Vukovic V., Cabarkapa M., Veseljević Jerkovic S., Jovančevic A. Evaluating large language models against human annotators in latent content analysis: Sentiment, political leaning, emotional intensity, and sarcasm. 2025. URL: https://doi.org/10.48550/arxiv.2501.02532 (Дата звернення: 23 березня 2025).

4. Feigel L. The Murder of Rosa Luxemburg review – tragedy and farce. The Guardian. 2019. URL: https://www.theguardian.com/books/2019/jan/09/the-murder-of-rosa-luxemburg-by-klaus-gietinger-review (Дата звернення: 21 квітня 2025).

5. Gole M., Nwadiugwu W.-P., Miranskyy A. On sarcasm detection with OpenAI GPT-based models. 2023. URL: https://doi.org/10.48550/arXiv.2312.04642 (Дата звернення: 23 березня 2025).

6. Matloga P., Marivate V., Olaleye K. Sentiment analysis using unsupervised learning for local government elections in South Africa. JeDEM – eJournal of eDemocracy and Open Government. 2025. Vol. 17, No. 1. P. 144–169. DOI: https://doi.org/10.29379/jedem.v17i1.945 (Дата звернення: 30 березня 2025).

7. OpenAI. API pricing. URL: https://openai.com/api/pricing/ (Дата звернення: 10 квітня 2025).

8. OpenAI. Introducing OpenAI o3 and o4-mini. 2025. URL: https://openai.com/index/introducing-o3-and-o4-mini/ (Дата звернення: 10 квітня 2025).

9. OpenAI. Structured outputs. URL: https://platform.openai.com/docs/guides/structured-outputs (Дата звернення: 10 квітня 2025).

10. Ornstein J. B., Blasingame A., Truscott J. B. How to Train Your Stochastic Parrot: Large Language Models for Political Texts. 2022. URL: https://joeornstein.github.io/publications/ornstein-blasingame-truscott.pdf (Дата звернення: 23 березня 2025).

11. Telegram Messenger Inc. Telegram Database Library (TDLib). URL: https://core.telegram.org/tdlib (Дата звернення: 8 листопада 2024).

12. wiz0u. WTelegramClient[Source code]. 2025. URL: https://github.com/wiz0u/WTelegramClient (Дата звернення: 30 березня 2025).