Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка
Навчально-науковий інститут філології
Кафедра української мови та прикладної лінгвістики

Автоматичний тональний аналіз українськомовних текстів новин

Кваліфікаційна робота
освітнього ступеня «бакалавр»
студентки 4 курсу
освітньої програми
«Прикладна (комп'ютерна) лінгвістика
та англійська мова»,
спеціальності – 035 Філологія (035.10 Прикладна лінгвістика)
галузі знань – 03 гуманітарні науки
Аліна Сергіївна СИНИЦЬКА
Науковий керівник:
Валентина РОБЕЙКО

«Допущено до захисту»
Протокол засідання
кафедри української мови та прикладної лінгвістики
від «06» 06 2024 року № 15

завідувач кафедри _____к.філол.н., доц. Сергій РІЗНИК

КИЇВ – 2024

Анотація

Актуальність дослідження обумовлена зростаючим обсягом текстових даних у соціальних мережах, які містять важливу інформацію про громадську думку та настрої. Об’єктом дослідження є тексти повідомлень українською мовою з мережі «Telegram», а предметом — методи та алгоритми автоматизованого визначення лінгвістичних ознак тональності.

Метою кваліфікаційної роботи є створення комп’ютерної системи для автоматичного визначення тональності текстів. Для досягнення цієї мети були виконані завдання з опрацювання літератури, аналізу існуючих методів тонального аналізу, опрацювання корпусу текстів новин українською мовою, тренування різних моделей для сентимент-аналізу та оцінки результатів роботи.

Методологія дослідження базується на застосуванні сучасних алгоритмів машинного навчання, що дозволяє ефективно аналізувати тексти в умовах динамічних змін інформаційного простору. Використання алгоритмів обробки природної мови, зокрема для української мови, забезпечує високу точність класифікації тональності текстів. Новизна дослідження полягає у застосуванні цих методів до українських текстів, що дозволяє адаптувати підходи тонального аналізу до специфіки національного інформаційного контенту.

У результаті проведеного дослідження було натреновано чотири різні моделі: визначення тональності з використанням тонального словника української мови, визначення тональності за ембедингами слів, що згенеровані локально розгорнутою мовною моделлю BERT, визначення тональності з попередньо натренованою моделлю Gemini API та з використанням попередньо навченої трансформерної моделі RoBERTa. Моделі Gemini API та RoBERTa показали найкращі результати: для Gemini API точність — 0.772 та для другої моделі точність — від 0.593 до 0.645, та мітка F1 — від 0.590 до 0.641. На основі останньої моделі було створено вебзастосунок для зручності тестування. Розроблена система дозволяє швидко та об'єктивно аналізувати тональність новинних матеріалів, що важливо для моніторингу громадської думки та виявлення фейкових новин та пропаганди.

Ключові слова: тональний аналіз, машинне навчання, трансформери, українські тексти, автоматизація, великі мовні моделі.

Abstract

The relevance of the study is due to the growing amount of textual data in social media, which contains important information about public opinion and sentiment. The object of the study is the texts of messages in Ukrainian from the Telegram network, and the subject is the methods and algorithms for automated detection of linguistic features of tone.

The aim of the qualification work is to create a computer system for automatically determining the tone of texts. To achieve this goal, the tasks of literature research, analysis of existing methods of tone analysis, processing of the corpus of news texts in Ukrainian, training of various models for sentiment analysis and evaluation of the results of work were performed.

The research methodology is based on the use of modern machine learning algorithms, which allows for effective text analysis in the context of dynamic changes in the information space. The use of natural language processing algorithms, in particular for the Ukrainian language, ensures high accuracy of text tone classification. The novelty of the study lies in the application of these methods to Ukrainian texts, which allows us to adapt the approaches of tone analysis to the specifics of national information content.

As a result of the study, four different models were trained: pitch detection using the Ukrainian tone dictionary, pitch detection based on word embeddings generated by the locally deployed BERT language model, pitch detection with the pre-trained Gemini API model and using the pre-trained RoBERTa transformer model. The Gemini API and RoBERTa models showed the best results: for Gemini API, the accuracy was 0.772, and for the second model, the accuracy ranged from 0.593 to 0.645, and the F1 label from 0.590 to 0.641. Based on the latter model, a web application was created to facilitate testing. The developed system allows for a quick and objective analysis of the tone of news stories, which is important for monitoring public opinion and identifying fake news and propaganda.

Keywords: tonal analysis, machine learning, transformers, Ukrainian texts, automation, large language models.

ЗМІСТ

Вступ

У сучасному цифровому світі, де інформаційні потоки стають все більш інтенсивними, аналіз тональності текстів набуває особливого значення. Соціальні мережі, такі як “Twitter” та “Telegram”, стали важливими платформами для обміну думками, новинами та інформацією. Автоматичний тональний аналіз цих текстів дозволяє швидко та ефективно виявляти емоційне забарвлення публікацій, що є важливим для моніторингу громадської думки, оцінки реакцій на події, а також для виявлення фейкових новин та пропаганди.

Ця кваліфікаційна робота присвячена автоматичному тональному аналізу текстів новин із Telegram-каналу. **Об'єктом дослідження** є тексти повідомлень українською мовою, опубліковані в соціальній мережі “Telegram”, а саме в новинних каналах. Ці тексти вирізняються постійними змінами та різноманіттям, що ускладнює їхній аналіз, але водночас робить його надзвичайно важливим для розуміння поточних соціальних та інформаційних процесів. **Предметом дослідження** є методи та алгоритми автоматизованого визначення тональності текстів, зокрема текстів новин українською мовою, а також лінгвістичні ознаки, які ми можемо виявити за допомогою цих методів. Це включає в себе збір текстових даних, їхню попередню обробку, аналіз та класифікацію за тональністю (позитивна, негативна, нейтральна).

Метою цієї роботи є створення комп’ютерної системи для автоматичного визначення тональності текстів. Завдяки розробленій системі, дослідники, маркетологи, політики та інші зацікавлені сторони зможуть швидко та об'єктивно аналізувати тон новинних матеріалів. Це допоможе моніторити громадську думку, оцінювати реакцію на події та продукти, а також приймати рішення на основі зібраної інформації. Система буде корисною для аналізу інформаційного простору, виявлення пропаганди, фейкових новин та інших маніпулятивних технік.

Дослідження **актуальне**, оскільки сучасне інформаційне суспільство характеризується величезним обсягом текстових даних, які надходять з різних джерел, зокрема соціальних мереж і месенджерів. Ці дані містять важливу

інформацію про настрої та думки суспільства, які можуть бути використані для різних цілей, від маркетингових кампаній до політичних стратегій. Зокрема, у контексті українського інформаційного простору, новини та повідомлення з Telegram-каналів набули значного впливу. В умовах гібридної війни, де інформаційні атаки та пропаганда є поширеними, можливість оперативно аналізувати тональність текстів стає критично важливою для виявлення маніпуляцій та пропагандистських наративів.

Для досягнення мети дослідження необхідно вирішити такі **завдання**:

1) Опрацювати літературу з основних понять, таких як тональність, думка та суб'єктивність.

2) Визначити основні проблеми у машинному тональному аналізі.

3) Проаналізувати існуючі методи автоматичного тонального аналізу.

4) Опрацювати незбалансований корпус текстів новин українською мовою.

5) Натренувати модель для сентимент-аналізу на корпусі текстів.

6) Оцінити результати роботи моделі та зробити загальні висновки щодо проведеного дослідження.

Структура й обсяг кваліфікаційної роботи. Робота складається зі вступу, трьох розділів: теоретичної частини (Розділ 1. Автоматичний тональний аналіз: огляд наукових джерел; Розділ 2. Огляд існуючих методів аналізу тональності текстів та практичних результатів (Розділ 3. Обробка текстів новин з Telegram: автоматизоване визначення тональності), містить висновки після кожного розділу та загальні висновки, список використаних джерел, додатки. Загальний обсяг кваліфікаційної роботи становить 63 сторінки. Із них основного тексту 51 сторінка, список використаних джерел (63 найменування) – на 6 сторінках та додатки на 1 сторінці.

РОЗДІЛ 1. Автоматичний тональний аналіз: огляд наукових джерел

У цьому розділі ми проаналізуємо основні поняття та завдання галузі автоматичного тонального аналізу текстів, розглянемо важливі аспекти, які впливають на точність та ефективність такого аналізу, а також опишемо основні проблеми, з якими стикаються дослідники у цій сфері.

1.1. Основні поняття галузі автоматичного тонального аналізу текстів

Аналіз тональності текстів (англ. Sentiment Analysis) — це область дослідження, що включає процес визначення емоційного забарвлення або настрою, висловленого в тексті. Таким чином, головна мета такого аналізу полягає в тому, щоб визначити, наскільки позитивним, негативним або нейтральним є висловлювання в тексті.

Сентимент-аналіз, відомий також як емоційний аналіз тексту, є областю обробки природних мов, що спрямована на автоматичне визначення та класифікацію емоційного відтінку в текстових даних. Це досягається за допомогою аналізу лексичних одиниць, синтаксичних структур, контексту та інших факторів, що вказують емоційний аспект. Наприклад, в аналізі лексичних одиниць можуть враховуватись слова з відомою емоційною забарвленістю, такі як “радісний”, “сумний” та “злісний”. Контекстуальний аналіз зазвичай включає розглядання контексту, у якому текст з'явився, оскільки це може визначати емоційні нюанси, що не виражені прямо в тексті. Використання метафор, сарказму, епітетів та інших лінгвістичних прийомів також впливають на емоційне сприйняття тексту.

У наш час аналіз тональності текстів часто використовується в різних галузях, таких як маркетинг, фінанси, соціальні науки та багато інших. Він допомагає розуміти реакції аудиторії на різні події, продукти чи послуги, а також дозволяє швидко аналізувати великі обсяги текстів для отримання об'єктивних висновків про їхню емоційну природу. Ресурсами для тонального аналізу текстів можуть стати [17] блоги, форуми, новинні статті, соціальні

мережі (такі як “Twitter”, “Telegram”, “Facebook”, “Reddit”) та відгуки на товари та послуги.

Аналізуючи велику кількість літератури, було помічено, що багато хто з дослідників поєднує поняття — аналіз думок (англ. Opinion Mining) і аналіз тональності (англ. Sentiment Analysis), з тієї причини, що перший є відгалудженням сентимент-аналізу, тому їх сприймають як одну задачу. Вказуючи, що вони є взаємозамінними, тобто це процес комп'ютерного вивчення поглядів, відношень і емоцій, які люди мають до певної теми чи об'єкта. Попри це, є і ті, хто розмежовує ці начебто близькі поняття. А вбачають вони різницю в тому, що аналіз думок виділяє й аналізує думку людей про об’єкт, а тональний аналіз визначає почуття, виражені в тексті, а потім аналізує його. Тому метою другого є пошук думок, ідентифікація настроїв, які вони виражають, а потім класифікація їх полярності [13].

З точки зору дослідника Бінг Лю думка складається з п‘ятьох компонентів: <e, a, s, h, t> [12]:

- e — entity (сутність або об'єкт) те, що оцінюється чи описується у думці. Наприклад, конкретний продукт, послуга, місце, подія або будь-який інший об'єкт.
- a — aspect (аспект або ознака) — конкретний аспект або характеристика сутності, яка оцінюється у думці. Наприклад, якщо сутність — це ресторан, аспекти можуть включати якість їжі, обслуговування, атмосферу тощо.
- s — sentiment (настрій чи оцінка) — емоційний стан або оцінка, що виражається у думці. Він може бути позитивним, негативним або нейтральним, а також може виражати різні градації цих настроїв.
- h — holder (власник думки) — особа чи група, які висловлюють думку чи оцінку. Це може бути конкретна людина, колектив або взагалі не вказано.
- t — time (час висловлення) — момент чи період, коли була висловлена думка. Час може бути точно визначений (наприклад, дата та час) або загальний (наприклад, «цього літа», «у 2023 році»).

Наведемо приклад відгуку, де матимемо такі аспекти:

1. e — сутність: Ароматична веганська свічка.
2. a — ознака: Запах свічки.
3. s — оцінка: Погана. (2 зірки з 5)
4. h — власник думки: Марина.
5. t — час висловлення: 13 березня 2024 року.

Приклад відгуку на аромасвічку з сайту [23]:

Марина ★★☆☆☆ 13 березня 2024

Покупку підтверджено Запах неприємний, дуже “хімозний.”

Зазначимо, що думки поділяються на два типи — пряма думка (англ. Direct opinion), коли людина висловлює свою думку про певну ознаку об'єкта. Наприклад, вона може сказати, що подорож була цікавою (позитивна думка), книжка нудна (негативна думка) або нейтрально висловити своє ставлення до якоїсь ознаки. Для цього вона може використовувати різні слова або фрази, щоб описати цю ознаку. Та порівняльна думка (англ. Comparative opinion) висловлює те, як об'єкти подібні чи відрізняються між собою. Це може бути також вибір об'єкта згідно з певними ознаками. Наприклад, коли ми порівнюємо дві різні книги та обираємо ту, де цікавіший сюжет. Зазвичай порівняльну думку виражають за допомогою слів, які показують рівень якості, наприклад, «смачніший», «швидший» тощо.

Іншою класифікацією думок є поділ їх на явні та неявні. Явна думка — це суб’єктивне твердження, яке дає звичайну або порівняльну думку. Наприклад, «Кока-кола чудова на смак» і «Кола-кола смачніша за пепсі». Тоді як неявна думка — це об'єктивне твердження, яке передбачає звичайну або порівняльну думку. Таке об’єктивне твердження зазвичай виражає бажаний чи небажаний факт, наприклад, «Я купив матрац тиждень тому, і він змінив форму» або «Заряд батареї телефонів Nokia довший, ніж телефонів Samsung». [12].

Згідно з працею «Learning Subjective Language» [10] суб'єктивність та об'єктивність є ключовими концепціями у темі тонального аналізу текстів. Вони

вказують на різні підходи до вираження думок, емоцій та оцінок у тексті. Суб'єктивність — це вираження особистих думок, почуттів, оцінок або поглядів автора тексту. Суб'єктивні висловлювання зазвичай зображують емоційне забарвлення та ставлення до певної теми. Наприклад, «Цей фільм був захоплюючим і цікавим!» — суб'єктивне висловлювання про позитивне враження від фільму. «Мені не подобається смак цього напою» — суб'єктивне висловлювання про негативне ставлення до смаку напою. У свою чергу об'єктивність — це вираження фактів, без урахування особистих емоцій або оцінок. Об'єктивні висловлювання мають більш нейтральний характер і ставляться до предмету без особистого впливу автора. Наприклад: «У цьому тексті описано процес виробництва нового продукту» – об'єктивне висловлювання про фактичну інформацію. «Столиця країни розташована на північному заході» — об'єктивне висловлювання про географічне розташування.

Можемо підсумувати, що при аналізі тональності текстів, суб'єктивні висловлювання можуть містити емоційні слова або відзначати позитивний та негативний настрій, тоді як об'єктивні висловлювання скоріше фокусуються на фактах і не виражають особистих думок. За допомогою цих понять можна краще розуміти та аналізувати емоційну та об'єктивну сторони текстів у рамках тонального аналізу.

Наступним важливим для розуміння тонального аналізу текстів є його рівні, які можуть бути описані як діапазон від загального до деталізованого вивчення вираженого в тексті настрою або оцінки. У книзі «Sentiment Analysis and Opinion Mining» Б. Лю [12] виокремлює такі рівні:

1. Рівень документу — головною задачею є класифікація цілого документу (наприклад, статті, відгуку, огляду і т.д.) за позитивним, негативним або нейтральним настроєм. При цьому допускається, що документ виражає думки тільки щодо одного об'єкта або сутності.
2. Рівень речення — завдання полягає у перевірці кожного речення в тексті на вираження позитивного, негативного чи нейтрального відношення. Цей рівень аналізу тісно пов'язаний з класифікацією суб'єктивності, яка

відрізняє об'єктивні речення, що виражають фактичну інформацію, від суб'єктивних, що виражають особисті думки та погляди. Нейтральне відношення зазвичай означає відсутність думки.

3. Рівень аспектів — проводить більш деталізований аналіз, спрямований на виявлення того, що саме сподобалося або не сподобалося людям у тексті. Замість вивчення мовних конструкцій (документів, абзаців, речень, фраз або частин), рівень аспектів досліджує безпосередньо виражені в тексті оцінки, думки та погляди.

Кожен з цих рівнів аналізу відіграє важливу роль у розумінні тональності текстів. Від рівня документу, який надає загальний настрій документа, до рівня аспектів, який вивчає конкретні елементи, які викликали певні емоції або думки, кожен з них вносить свій внесок у вивчення вираженого в тексті відношення.

1.2. Основні завдання аналізу тональності

Розглянемо модель сутності. Припустимо, що сутність e_i представляється скінченним набором аспектів $A_i = \{a_{i1}, a_{i2}, \ldots\ldots a_{in}\}$. E_i може бути виражена будь-яким іншим набором виразів сутності $\{e_{i1}, e_{i2}, \ldots\ldots e_{in}\}$. Кожен аспект сутності може бути представлений як набір виразів $\{ae_{i1}, ae_{i2}, \ldots\ldots ae_{in}\}$[12].

Тобто виходить, що ми маємо документ d, що містить думки щодо набору об’єктів $\{e_1, e_2, \ldots\ldots e_n\}$ та їх аспектів від певного набору власників думки в певний період часу.

Для проведення аналізу настроїв на основі емоційно забарвленого документу D, необхідно виконати наступні 6 основних завдань:

Завдання 1. Вилучення сутностей та їх категоризація. Вилучити всі вирази сутностей у D та категоризувати їх, а також виконати категоризацію та розбиття їх на класи сутностей. Кожен клас характеризує окрему сутність.

Завдання 2. Вилучення аспектів і їх категоризація. Вилучити усі вирази аспектів сутностей у D і класифікувати ці вирази аспектів у класи сутностей. Кожен клас характеризує окрему сутність.

Завдання 3. Вилучення та категоризація власників думки. Вилучити власників думок із текстових або структурованих даних і класифікувати їх. Завдання аналогічне двом наведеним вище.

Завдання 4. Вилучення часу та стандартизація. Вилучити час, коли були висловлені думки та стандартизувати різні формати представлення часу. Завдання також аналогічне наведеним вище.

Завдання 5: Класифікація настрою за аспектами. Визначити, чи є думка щодо аспекту позитивною, негативною чи нейтральною, або призначити числовий рейтинг настрою аспекту.

Завдання 6: Створити усі п’ять елементів думки, що містять сутність, аспект, оцінку, власника та час вираження, на основі попередніх обговорень.

Аналіз настроїв на основі цих завдань часто називається аспектно-орієнтованим аналізом настроїв або аналізом настроїв на основі ознак [12].

1.3. Основні проблеми у галузі автоматичного тонального аналізу текстів

На перший погляд здійснення автоматичного тонального аналізу текстів не має викликати проблем. Але поринувши в цю сферу діяльності, ми з‘ясували, що в роботі можуть виникати різні перешкоди, що ускладнюють або спотворюють результати аналізу. Розберемо детальніше кожну проблему, яку вдалося віднайти у різних наукових джерелах [12, 14, 15, 57].

Контекст (або ж амбівалентність). Розуміння контексту є ключовим для правильної класифікації тональності, проте машини можуть мати проблеми з адекватним розумінням контексту, а це в свою чергу може призводити до неправильної класифікації. Проаналізуємо такий текст: «Їжа була досить смачна, але обслуговування було повільним і непривітним». Машина, що аналізує сентимент, сприйматиме слово «смачна» як позитивне, але без

контексту відгуку може неправильно класифікувати загальний сентимент як позитивний.

Сарказм та іронія. У деяких випадках люди використовують сарказм та іронію, коли вони висловлюють свої думки. Це може призводити до неправильного розуміння висловлювання і неправильної класифікації настрою. Наприклад, «Це дивовижна кавова машинка! Зламалася вже на третій день користування».

Неоднозначність. Слова або фрази можуть мати декілька значень або бути неоднозначними з точки зору тональності. Наприклад, залежно від контексту слово «молодий» може мати як позитивне — «Він молодий і енергійний, має багато ідей» , так і негативне значення — «Він ще надто молодий і недосвідчений для цієї посади»,.

Різноманітність виражальних засобів. Оскільки мова є дуже різноманітною, то висловлювання можуть бути виражені різними способами, використовуючи різні лексичні одиниці. Це в свою чергу ускладнює процес аналізу та потребує більш глибокого розуміння мови.

Невідомі слова та вирази. Якщо система не має достатнього словника або бази знань для розпізнавання нових слів або виразів, це може призвести до неправильної або неповної класифікації тональності. Тому бази даних та словники мають постійно оновлюватись.

Порівняння. Вирази, що використовують порівняння, можуть бути неоднозначними без знання того, з чим порівнюються елементи. Машинам може бути важко визначити, чи є порівняння позитивним, негативним чи нейтральним без контексту (див. пункт 1.1).

Емоджі. Емоджі можуть створювати проблеми при аналізі, особливо при роботі з соціальними медіа. Машини повинні правильно розпізнавати та класифікувати значення емоджі для якісного результату. Варто також зауважити, що емоджі можуть мати протилежний зміст, ніж текстове повідомлення, бо саме вони допомагають виразити сарказм та іронію у повідомленнях.

Ідіоми. Вживання ідіом може бути заплутаним для машин, які можуть неправильно інтерпретувати такі вирази без відповідного навчання.

Нейтральність. Правильне визначення нейтральних відгуків є важливим для точного аналізу сентименту. Машини повинні розпізнавати об'єктивні висловлювання, щоб не класифікувати їх як позитивні або негативні.

Негація. Це ускладнює роботу моделям аналізу тональності, оскільки може змінювати смисл висловленого у реченні сентименту. Наприклад, у фразі «Ця кава не є поганою» негація знімається, і вираз насправді виражає позитивний намір. Для правильного розуміння таких конструкцій машини повинні бути навчені розрізняти, як кілька негацій можуть анулювати одна одну.

Риторичні запитання. Риторичні запитання можуть сприйматися як прості запити без емоційного відтінку або як висловлювання, що виражають емоційний стан мовця.

Ці аспекти показують складність завдання автоматичного аналізу тональності та необхідність постійного вдосконалення та розвитку систем, що виконують цю задачу.

Висновки до розділу 1

Проведений огляд літератури щодо аналізу тональності текстів розширив наше розуміння того, як емоційне забарвлення впливає на сприйняття інформації.

Ми з‘ясували, що думка складається з 5 елементів: сутність, аспект, настрій, власник думки та час висловлення. Також дізналися, що думки можуть класифікуватися на прямі/порівняльні та явні/неявні. Цікаво було аналізувати та досліджувати різноманітні проблеми, з якими може стикатися автоматичний тональний аналіз. Основні завдання аналізу тональності та їх деталізація в огляді допомагають краще зрозуміти різні етапи обробки та аналізу текстових даних для вилучення думок та оцінок, що стане у пригоді для подальшої практичної реалізації.

Найбільш корисними працями для нашої подальшої роботи є «Sentiment Analysis and Opinion Mining» Б.Лю, «Challenges in Sentiment Analysi» Саїф М.

Мохаммад та «Opinion mining and sentiment analysis» Бо Панг і Ліліан Лі та інші. Із них ми отримали цінну інформацію про основні поняття, завдання, проблеми та підходи в галузі аналізу тональності текстів.

Огляд надав достатнє вступне уявлення про галузь аналізу тональності текстів, її основні концепції, завдання та проблеми. Це дозволить краще орієнтуватися в цій темі при вирішенні практичних завдань.

РОЗДІЛ 2. Огляд існуючих методів аналізу тональності текстів

При роботі з автоматичним тональним аналізом ми можемо використати три основні підходи — підхід на основі лексиконів (англ. Lexicon Based Approach), підхід на основі машинного навчання (англ. Machine Learning Approach) та гібридний підхід (англ. Hybrid Approach), кожен з яких детальніше розглянемо у цьому розділі.

Малюнок 2.1. Методи аналізу тональності текстів [18]

2.1. Підхід на основі лексиконів для сентимент-аналізу

Lexicon Based Approach. Лексикони — це набір лексем, де кожній лексемі присвоюється заздалегідь визначена оцінка, яка вказує на нейтральний, позитивний чи негативний характер тексту. Оцінка присвоюється лексемам на основі полярності, наприклад, + 1, 0, - 1 для позитивного, нейтрального, негативного, або оцінка може бути присвоєна на основі інтенсивності полярності. Оскільки навчальні дані не потрібні, цей метод можна назвати неконтрольованим (англ. unsupervised) [21].

Головними плюсами такого підходу є: легкість у використанні та розумінні, так як метод простий у застосуванніі і не потребує спеціалізованих знань для інтерпретації результатів. Швидкість аналізу, тому що даний підхід забезпечує швидке виявлення емоційного забарвлення тексту без великих обчислювальних витрат. І останній виокремлений пункт це доступність ресурсів — лексикони та словники, що використовуються у цьому підході, зазвичай доступні для використання безкоштовно або за невелику плату.

Незважаючи на всі плюси, також є значні мінуси: обмежена здатність у розрізненні контексту, підхід може не завжди точно відтворити емоційний тон тексту, оскільки не ураховує контекстуальні відтінки слів. Схожим мінусом до першого є брак гнучкості — підхід обмежений у врахуванні синтаксичної та семантичної інформації, що може впливати на точність аналізу. А також залежність від якості джерела даних, тобто результати аналізу суттєво залежать від якості та охопленості використаного лексикону чи словника, що може впливати на точність виявлення емоційного забарвлення тексту. Це підкреслює важливість наявності експертів у галузі, які мають досвід у створенні та підтримці лексиконів для аналізу тексту. Однак цей процес може бути дуже затратним та часомістким.

У рамках лексичного підходу можна виділити два основних пункти: підхід на основі словника (англ. Dictionary Based Approach), а також корпусний підхід (англ. Corpus Based).

Підхід на основі словника (англ. Dictionary Based Approach) — у цьому випадку основним ресурсом виступає словник слів та виразів, які виражають певний сентимент або думку. Цей словник формується вручну, шляхом додавання слів з позначенням їх полярності — позитивної, негативної чи нейтральної.

У контексті цього можемо згадати тональний словник української мови Оксани Толочко [47], який був створений нещодавно, що забезпечує його актуальність та відповідність сучасним мовним тенденціям. Дані зберігалися у форматі CSV, що полегшує їхнє використання у автоматичному визначенні

настрою текстів. Для визначення тональності використовувалась така шкала: 2 — дуже позитивно, 1 — позитивно, 0 — нейтрально, -1 — негативно, -2 — дуже негативно.

Іншим підходом є корпусний підхід (англ. Corpus Based), який використовує семантичні та синтаксичні шаблони для виявлення емоційно забарвлених слів та визначення їх полярності в реченнях.

Статистичний підхід у сентимент аналізі використовує статистичні методи для виявлення закономірностей в текстах, аналізуючи, як часто певні слова або фрази зустрічаються у позитивних, негативних або нейтральних контекстах і на основі цього визначаючи їх емоційне забарвлення.

Семантичний підхід, з іншого боку, зосереджується на значенні слів та їх взаємозв'язках. Він використовує ресурси (WordNet, SentiWordnet), які містять інформацію про значення слів, щоб визначити, наскільки схожі слова та як вони відображають емоційні аспекти. Наприклад, якщо слова «радісний» і «щасливий» мають близьке значення, то їх полярність вважається схожою.

Таким чином комбінування цих підходів може допомогти отримати більш повний та точний аналіз емоційного забарвлення текстів, оскільки вони доповнюють один одного і дають можливість розглядати проблему з різних сторін.

2.2. Підхід на основі машинного навчання для сентимент-аналізу

Machine Learning Approach. Підхід на основі машинного навчання в автоматичному тональному аналізі текстів використовує комп'ютерні алгоритми та моделі для виявлення емоційного забарвлення текстів. Основна ідея полягає у тому, щоб навчити комп'ютерну систему розрізняти та класифікувати тексти за їхньою емоційною спрямованістю на позитивну, негативну або нейтральну.

Алгоритми, що базуються на машинному навчанні, навчають класифікатор на основі даних, розмічених вручну. Однак, якість та охоплення навчальних даних мають великий вплив на продуктивність класифікатора, тобто

він потребує великої бази даних для ефективної роботи, що є його основним недоліком. Цей підхід має кращу точність, ніж лексикографічний [18] .

Під час навчання моделі використовуються текстові дані разом з мітками, що позначають емоційну спрямованість кожного тексту. Після навчання модель може класифікувати нові тексти за їхньою емоційною спрямованістю з високою точністю, використовуючи вивчені закономірності та шаблони в текстах.

Також потрібно детальніше розібрати плюси та мінуси такого підходу. Найголовнішим плюсом є висока точність у визначенні емоційного забарвлення текстів, особливо при належному підборі навчальних даних і параметрів моделі. Крім того моделі, які пройшли навчання за допомогою машинного навчання, можуть автоматично адаптуватися до нових даних і умов, що дозволяє їм зберігати високу ефективність у реальному часі. Щодо застосування, то цей підхід можна успішно використовувати для аналізу різноманітних текстів: соціальних мереж, новинних порталів, оглядів товарів і послуг тощо.

Тепер розглянемо детальніше недоліки, а саме потребу у великій кількості даних, так як для успішного навчання моделі необхідно мати достатньо великий та репрезентативний набір навчальних даних, що може бути витратним і складним завданням. Питання обробки неструктурованих даних, так як тексти є формою неструктурованих даних, що може призвести до викликів при їх обробці та аналізі за допомогою машинного навчання. Важливим є також те, що навчання моделі вимагає експертного знання у галузі, щоб правильно обрати параметри моделі, оптимізувати її та врахувати можливі впливи факторів на результат. І останній недолік — моделі можуть мати складність у визначенні емоційного забарвлення в тексті, особливо якщо контекст емоцій змінюється або неоднозначний. Наприклад, якщо в одному реченні слово вживається в позитивному контексті, а в іншому — у негативному, це може призвести до плутанини в аналізі для моделі.

Машинне навчання включає в себе два підходи — контрольоване машинне навчання (англ. Supervised machine learning) та неконтрольоване навчання на основі лексики (англ. Lexicon-based unsupervised learning) [21].

Неконтрольовані стратегії аналізу настроїв використовують бази знань, онтології, бази даних і лексикони, які містять детальні знання, відібрані та підготовлені спеціально для аналізу настроїв. Методи керованого навчання є більш поширеними завдяки їхнім точним результатам. Ці алгоритми повинні бути навчені на навчальній вибірці, перш ніж їх можна буде застосувати до реальних даних [21].

Проаналізуємо контрольоване машинне навчання та кілька його типів — ймовірнісний класифікатор, лінійний класифікатор та дерево рішень.

Ймовірнісний класифікатор (англ. Probalistic classifier) використовує змішані моделі для класифікації. Модель змішаного розподілу передбачає, що кожний клас є компонентом змішаної моделі. Кожен компонент змішаної моделі є генеративною моделлю, яка надає ймовірність вибору певного терміна для цього компонента. Ці види класифікаторів також називають генеративними класифікаторами [22]. Три відомі ймовірнісні класифікатори — це наївний Байєс, байєсівська мережа та класифікатор максимальної ентропії.

1. Звичайний наївний Баєсів класифікатор (англ. Naïve Bayes Classifier; NB) — простий, але ефективний ймовірнісний класифікатор, який ґрунтується на теоремі Баєса. Основна його ідея полягає у тому, що він вважає всі ознаки незалежними між собою, незалежно від класу. Це спрощення дозволяє швидко та ефективно навчати модель та здійснювати класифікацію. NB часто використовується для текстової класифікації, де кожне слово або ознака може розглядатися як незалежний атрибут.

Формула теореми Баєса [35] — $P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$

NB моделює ймовірності належності екземпляра до кожного класу на основі ознак. Для кожного класу розраховується ймовірність наявності певних ознак, а потім ці ймовірності знаходяться за допомогою теореми Баєса. Після того як були обчислені ймовірності для кожного класу, вхідний екземпляр класифікується як той клас, для якого ймовірність наявності вхідних ознак найвища.

2. Мережа Баєса (англ. Bayesian Network; BN) — цей алгоритм працює з припущенням, що всі ознаки незалежні одна від одної або повністю залежать одна від одної. Це наближається до моделі мережі Баєса — графа, в якому випадкові величини є вузлами, а умовні залежност і — ребрами. Оскільки складність мережі Баєса дуже висока, вона зазвичай не використовується [5].

3. Останній класифікатор — максимальної ентропії (англ. Maximum Entropy Classifiers; ME), що намагається зберегти якомога більше невизначеності. Обчислюються моделі, які відповідають навчальним прикладам, де кожна ознака відповідає обмеженню на модель. Для класифікації обирається модель з максимальною ентропією серед усіх моделей, які задовольняють ці обмеження [6]. Його перевагою є здатність працювати з різними типами ознак (категоріальними, числовими тощо) і здатність враховувати умовні залежності між ознаками у даних для кращої точності класифікації.

Лінійний класифікатор (англ. Linear classifier) — це клас алгоритмів машинного навчання, який використовує лінійні функції для розділення простору ознак на класи. Основна ідея полягає в тому, щоб знайти лінійну комбінацію ознак, яка найкраще відділить один клас від іншого. Розглянемо такі приклади класифікаторів - метод опорних векторів (SVM) та нейронні мережі.

1. Метод опорних векторів (англ. Support Vector Machine, SVM) намагається знайти гіперплощину, представлену вектором, який розділяє позитивні та негативні навчальні вектори документів з максимальним відривом [19]. SVM є дуже ефективним для роботи з невеликими та середніми розмірами даних та для завдань, де важливо точно розділити класи з високою роздільною здатністю. Однак він може бути менш ефективним для великих наборів даних через велику обчислювальну складність та потребу у належному підборі гіперпараметрів.

2. Нейронна мережа (англ. Neural Network, NN) — це математична модель, яка імітує структуру та функціонування біологічних нейронних

мереж з метою вирішення різноманітних задач, таких як класифікація, регресія, прогнозування та генерація. В основі нейромереж лежать штучні нейрони, які об’єднуються в графові структури і передають сигнали один одному через ваги зв’язків. Завдяки процесу навчання, під час якого ваги та зміщення між нейронами оптимізуються, нейромережі стають здатними до виявлення закономірностей та залежностей у вхідних даних [27].

Нейронні мережі використовуються для навчання моделей, які можуть розпізнавати емоційні відтінки слів та фраз. Одним з популярних підходів є використання рекурентних нейронних мереж (анг. Recurrent Neural Networks, RNN) або їх вдосконалених версій, таких як довга короткочасна пам'ять (англ. Long Short-Term Memory, LSTM) або вентильні рекурентні вузли (англ. Gated Recurrent Unit, GRU). Ці архітектури дозволяють моделі пам'ятати попередні контексти, що є важливим для аналізу тексту. Також використовуються згорткові нейронні мережі (англ. Convolutional Neural Network, CNN), які добре справляються з виявленням локальних особливостей у тексті, таких як ключові слова або вирази, що вказують на емоційний настрій.

Зазвичай для тренування таких моделей використовуються набори даних, які містять текст та мітки тональності для кожного текстового фрагмента. Після навчання модель може аналізувати нові тексти та визначати їхні емоційні відтінки з високою точністю.

Метод дерев рішень (англ. Decision trees), що також називають деревами вирішальних правил, деревами класифікації і регресії. У найбільш простому вигляді дерево рішень — це спосіб показу правил в ієрархічній, послідовній структурі. Основа такої структури — відповіді «Так» або «Ні» на низку питань [3]. Ця ієрархічна структура, де кожен вузол представляє умову або питання, а кожен гілка вибір або результат цієї умови.

Дерева рішень можуть бути ефективними для сентимент-аналізу через їхню простоту в інтерпретації та можливість генерації правил, які можна легко

зрозуміти, однак можуть вимагати великої кількості даних для досягнення достатньої точності і уникнення перенавчання.

Крім того, у сучасному машинному навчанні виділяють два основні підходи: класичне машинне навчання та навчання з використанням нейромереж.

Класичне машинне навчання включає алгоритми, які базуються на статистичних методах і теорії оптимізації. Ці алгоритми, такі як регресія, дерева рішень, методи опорних векторів (англ. Support vector machines, SVM), наївний Баєс та інші, використовуються для різних задач класифікації, регресії та кластеризації. Вони добре працюють з невеликими та середніми наборами даних і мають перевагу в простоті та інтерпретованості.

Навчання з використанням нейромереж, особливо глибокого навчання, стало в останні роки дуже популярним завдяки його здатності обробляти великі обсяги даних та складні задачі. Нейромережі, зокрема глибокі нейронні мережі (анг. Deep neural network, DNN), згорткові нейронні мережі (англ. CNN) та рекурентні нейронні мережі (англ. RNN), показали якісні результати в таких областях, як розпізнавання зображень, обробка природної мови, ігри та багато інших.

Основними перевагами глибоких нейромереж є: здатність обробляти великі дані, адже нейромережі можуть навчатися на величезних наборах даних, що дозволяє їм витягувати складні залежності і патерни. Вони показують відмінні результати в задачах, де необхідна висока точність, а також здатні автоматично визначати важливі ознаки в даних, що мінімізує потребу в ручному підборі ознак.

У світі машинного навчання з'являються нові тенденції, які змінюють традиційні підходи до цієї технології. Однією з найпомітніших тенденцій є виділення машинного навчання на основі великих мовних моделей (англ. Large language model, LLM) в окремий тип. Це обумовлено тим, що ці моделі використовують зовсім інші принципи роботи порівняно з традиційними підходами, зокрема трансформери та методи навчання на великих наборах текстових даних. Вони здатні розуміти та генерувати людську мову на більш

глибокому рівні, що відкриває нові можливості для застосувань у різних сферах, таких як обробка природної мови, створення контенту та багато іншого.

Яскравими прикладами найсучасніших LLM є двонаправлені кодерні представлення з трансформаторів (англ. Bidirectional Encoder Representations from Transformers, BERT), генеративні попередньо навчені трансформатори (англ. Generative pre-trained transformer, GPT) і Gemini [9]. Вони тренуються на значно більших наборах даних, що включають тексти з різних джерел, таких як книги, статті, веб-сторінки. Також процес попереднього тренування моделей LLM займає значний час і ресурси. Модель навчається на величезному обсязі текстових даних без конкретного завдання, але після передтренування вона може бути додатково налаштована (fine-tuning) на специфічні завдання.

Отже, великі мовні моделі можуть виконувати широкий спектр завдань без спеціального навчання на кожне завдання, що робить їх універсальними інструментами для обробки тексту, генерації мови, перекладу тощо.

З огляду на практичну частину, про яку буде йти мова в розділі 3, варто більш детально описати BERT [58] (англ. Bidirectional Encoder Representations from Transformers). Це передова система машинного навчання для обробки природної мови, розроблена компанією Google. Вона базується на підходах машинного навчання, а саме на нейронних мережах та методах глибокого навчання. BERT є «глибоко двонаправленою» та набуває розширених інтерпретацій текстів, беручи до уваги правий і лівий контексти однаково. Цей метод використовується для навчання універсальних мовних моделей на великих масивах даних і вирішення завдань NLP [16].

В основі BERT лежить потужна нейромережева архітектура, відома як трансформери (англ. Transformers). Ця архітектура включає механізм самоуваги, що дозволяє моделі зважувати значення кожного слова на основі його контексту, як попереднього, так і наступного. Це усвідомлення контексту надає BERT здатність генерувати контекстуалізовані вставки слів, які є представленнями слів з урахуванням їхніх значень у реченнях [29].

Також у цьому контексті ми можемо звернути увау на використання ембедингів (анг. word embedding) — числових векторних репрезентацій слів чи токенів у тексті, адже вони є ключовими компонентами сучасних моделей обробки природної мови на основі глибокого навчання.

У BERT ембединги генеруються у такий спосіб:

1. Вхідні текстові дані токенізуються — розбиваються на окремі токени (слова, підслова, символи тощо).
2. Для кожного токену створюється початковий ембединг — вектор чисел фіксованої довжини, який кодує його початкові характеристики.
3. Далі ці початкові ембединги проходять через декілька шарів нейронної мережі трансформерів з механізмами самоуваги та уваги між токенами.
4. На виході мережі генеруються контекстні ембединги, які кодують не лише початкові характеристики токенів, але й їх контекст у реченні чи документі.

Контекстні ембединги містять багату семантичну та синтаксичну інформацію про токени — їх значення, зв'язки з іншими токенами, роль у реченні тощо. Саме ці векторні репрезентації є ключем до потужних можливостей BERT у розумінні природної мови.

Розглянемо плюси та мінуси BERT. Щодо плюсів то це однозначно — точність, універсальність, адже можна використовувати для різноманітних завдань NLP, що робить її дуже гнучкою та потужною моделлю. Інший важливий аспект це її ефективність та відкритість, тобто модель з відкритим кодом, що робить її доступною для дослідників і розробників у всьому світі. Мінусами є такі аспекти: перш за все, це складна модель, яка потребує значних обчислювальних ресурсів для навчання та використання. Цікавим факом є те, що BERT — це “чорна скринька”, що ускладнює розуміння того, як вона приймає свої рішення, а також може успадковувати упередження з даних, на яких вона навчається. Останнім недоліком є те, що BERT потребує великих обсягів текстових даних для навчання, що може бути недоступним для деяких дослідників і розробників.

Використання саме Bidirectional Encoder Representations from Transformers у реалізації практичної частини пов’язане з опрацюванням великої кількості літератури на цю тему, а саме такі роботи, де здійснювалась реалізація для зібраних корпусів українських текстів [1, 10], що мали високі результати навчання моделі.

Покращеною версією моделі BERT є RoBERTa (англ. Robustly optimized BERT approach) [51], що була розроблена дослідниками з Facebook AI. Ця мовна модель була створена для покращення продуктивності в задачах обробки природної мови (NLP).

Основні особливості та вдосконалення RoBERTa: модель була натренована на значно більшому корпусі текстів, що дозволяє моделі краще розуміти різноманітні мовні патерни та контексти. Модель навчається протягом більшої кількості ітерацій, використовуючи довші текстові послідовності під час навчання, що допомагає їй краще захоплювати контекст та залежності у текстах. RoBERTa не використовує завдання передбачення наступного речення, яке було частиною BERT, тому що це незначно впливає на продуктивність моделі. Модель оптимізована за допомогою ретельного підбору гіперпараметрів, таких як розмір пакетів даних (англ. batch size) та швидкість навчання (англ. learning rate), що дозволяє досягти кращих результатів без значного збільшення витрат на обчислення. І останній аспект це те, що RoBERTa використовує більші вхідні послідовності під час навчання (до 512 токенів), що дає можливість моделі захоплювати більш довгострокові залежності в текстах.

Також оглянемо детальніше модель Gemini [43] — це велика мовна модель (LLM) від компанії Google DeepMind, що була розроблена для покращення взаємодії між людьми та штучним інтелектом, включаючи чат-ботів, генерацію тексту, переклад та інші завдання. Gemini поставляється в трьох версіях, призначених для різних рівнів обчислювальних потужностей: Gemini Pro, Gemini Nano та Gemini Ultra [8].

Gemini Pro – це покращена велика мовна модель, створена для розуміння та генерації людської мови. Основні характеристики GPro включають наступні: узагальнення тексту, розпізнавання об'єктів, розуміння контенту, генерація контенту, екстраполяція та класифікація щодо тональності [8].

Використання моделі Gemini для тренування автоматичного тонального аналізу текстів має свої переваги через ряд факторів: модель навчається на великому обсязі даних, що дозволяє їй демонструвати високу точність у розпізнаванні тону текстів. Модель здатна утримувати контекст у довгих текстах та діалогах, що є важливим для правильного розпізнавання тональності. Також вона гнучка у налаштуванні, тому що може бути налаштована для різних галузей та завдань, а також може бути масштабована для обробки великих обсягів тексту та високих навантажень.

З огляду літератури на тему великих мовних моделей [7, 8, 11] ми з‘ясували, що використання моделі Gemini для тренування автоматичного тонального аналізу текстів може покращити якість результатів та забезпечити більш точне і надійне визначення тональності у текстових даних. Результати такого підходу будуть реалізовані у розділі 3.

2.3. Гібридний підхід для сентимент-аналізу

Гібридні методи для сентимент-аналізу поєднують у собі різні підходи та техніки з метою отримання більш точного та надійного результату. Вони можуть бути ефективними в ситуаціях, коли немає одного універсального методу, який би працював для всіх типів аналізованого тексту. Наведемо деякі приклади таких методів:

1. Комбінація правил та машинного навчання, тобто у цьому методі використовуються правила, щоб визначити емоційно насичені слова та фрази у тексті. Потім застосовується модель машинного навчання, така як нейронна мережа або SVM (Support Vector Machine), для класифікації тексту на позитивний, негативний або нейтральний, використовуючи ці правила як основу.

2. Використання ембедингів слів та моделей машинного навчання, таким чином ми можемо використовувати ембединги слів для врахування семантики та контексту слова у тексті. Наприклад, можна використовувати Word2Vec [45] або GloVe [48] для отримання векторних представлень слів. Після цього можна застосовувати моделі машинного навчання, такі як LSTM або Random Forest, для аналізу тексту та визначення його тональності.

3. І останнє – це комбінація машинного навчання та аналізу на основі доменних знань. У цьому методі можна поєднувати результати моделей машинного навчання з аналізом на основі доменних знань. Наприклад, можна використовувати навчену модель машинного навчання для визначення емоційної тональності, а потім застосовувати додаткові правила чи спеціалізовані словники для виправлення або покращення результатів, особливо в специфічних галузях, де важливі додаткові контекстуальні знання.

Висновки до розділу 2

У цьому розділі ми ознайомилися з різноманітними підходами та методами, які використовуються для аналізу тональності текстових даних. Важливим аспектом, який було зазначено під час огляду літератури, є виокремлення не тільки традиційних, а й нових тенденцій у сфері сентименту, зокрема використання нейронних мереж та великих мовних моделей.

На основі цього ми визначили, що в подальшій роботі слід детальніше розглянути конкретні моделі та алгоритми, які можна використовувати для аналізу тональності тексту в певному контексті, а саме використаємо три підходи — традиційний на основі словників, а також моделі машинного навчання BERT, RoBERTa і Gemini. Також важливо буде вивчити можливість комбінування різних методів для досягнення кращих результатів

Було розглянуто значну кількість літератури з різних джерел та досліджень, пов'язаних з методами аналізу тональності текстів. Цінними були роботи: «A survey on sentiment analysis methods, applications, and challenges» Ванкхаде М., Рао А. К. С., Кулкарні К., «Sentiment Analysis — Methods,

Applications & Challenges» Бхонд С. Б., Прасад Д. Р., «Аналіз тональності текстів українською мовою» Рябишев О., Єрохін А., Бахмет А., «ChatGPT vs Gemini vs LLaMA on Multilingual Sentiment Analysis» Бушемі А., Провербіо Д. та багато інших. Широкий огляд літератури став основою для глибшого розуміння проблематики та визначення стратегій подальшої роботи у даній галузі.

РОЗДІЛ 3. Обробка текстів новин з Telegram: автоматизоване визначення тональності

У цьому розділі ми проаналізуємо процес обробки українськомовних текстів новин з платформи “Telegram”. Основна увага буде приділена дослідженню та визначенню найефективнішого методу для автоматизованого визначення тональності цих текстів. Результати цього дослідження дозволять краще зрозуміти, як різні методи справляються із завданням аналізу тональності новинних текстів і які з них є найбільш точними та надійними.

3.1. Збір текстів новин з Telegram-каналів

У сучасному інформаційному просторі соціальні мережі відіграють важливу роль у розповсюдженні новин та інформації. Однією з найбільш популярних платформ в Україні є “Telegram” — месенджер, який поєднує функціонал соціальної мережі та зручні інструменти для обміну повідомленнями. З моменту свого запуску в 2013 році Telegram здобув широку популярність завдяки високому рівню безпеки, швидкості обміну повідомленнями та можливості створення каналів для широкої аудиторії.

Перевагою Telegram є його оперативність і зручність у поширенні і отриманні інформації, саме тому аудиторія вподобала цей додаток, що надає доступ до каналів швидких новин, стрічка яких формується за хронологічним принципом [4].

Останнім часом, особливо в умовах повномасштабної російсько-української війни, Telegram-канали набули особливого значення. Більшість з них мають інформаційний та новинний характер, надаючи користувачам швидкий доступ до важливої та актуальної інформації. В умовах війни, коли інформація може змінюватися дуже швидко, цей месенджер став невід'ємним інструментом для отримання оперативних новин та повідомлень про поточну ситуацію. Тому саме цю соціальну мережу було обрано для створення корпусу.

Для проведення навчальних досліджень у галузі автоматичного тонального аналізу текстів українською мовою Мураховською Вікторією [39] був зібраний корпус текстів новин з Telegram-каналів. Ці новинні тексти були позначені як позитивні, негативні або нейтральні з точки зору їх емоційного забарвлення та загального тонального відтінку.

Важливо відзначити, що наданий корпус новинних текстів був незбалансованим, тобто кількість повідомлень з різними тональностями була нерівномірною. Це відображає реальну ситуацію в новинному середовищі, де певні настрої можуть переважати залежно від тематики та контексту новин. Незбалансованість корпусу є однією з типових проблем, з якою доводиться стикатися в завданнях тонального аналізу текстів.

Ми ознайомились з сновними принципами роботи спеціального застосунку, що мав можливість завантажувати та обробляти новинні тексти з Telegram-каналів, а також класифікувати з точки зору тонального забарвлення.

Слід зазначити, що повний вихідний код цього застосунку залишається приватною власністю. Це пов'язано з конфіденційністю та захистом інтелектуальної власності в галузі технологій. Канали, для яких потрібно було зібрати тексти, були відібрані мною. Це дозволило зосередитися на конкретних джерелах інформації та забезпечити відповідність зібраних даних дослідницьким цілям.

У роботі були використані такі канали: «Forbes Ukraine» [41], «Бізнес України» [24], «Київ Оперативний|Kyiv Operative» [26], «Новинарня» [28], «Реальний Київ| Україна» [29], «Спорт України» [30], «ТСН новини/ТСН.ua» [32] та «Труха Україна» [31].

Основні принципи роботи аналізованого застосунку:

1) Для завантаження постів з Telegram-каналів у рамках дослідження використовувалися Telegram API [61] та стороння бібліотека Telethon [62]. Ці інструменти забезпечують ефективний доступ до контенту Telegram-каналів і автоматизують процес збору даних.

2) Для зберігання даних у цьому проєкті було обрано NoSQL бази даних (анг. Not Only SQL). Це група баз даних, які відрізняються від традиційних реляційних баз даних (SQL) способом організації, зберігання та доступом до даних. NoSQL бази даних зазвичай потрібні для зберігання великих обсягів неструктурованих або слабоструктурованих даних [42].

   Таким чином для зберігання та обробки даних було обрано Elastic (Elasticsearch). Це розподілений, RESTful пошуково-аналітичний рушій, який дає змогу зберігати, шукати й аналізувати великі обсяги даних практично в режимі реального часу [25]. Також ElasticSearch може бути інтегрований з Kibana, потужним інструментом для візуалізації та аналізу даних. Це дозволяє створювати інтерактивні дашборди, графіки та звіти для візуального аналізу даних, виявлення тенденцій та закономірностей.
3) Приписування тональності текстів здійснювалося за допомогою мовної моделі GPT-4-turbo, як найновішої з доступних Open AI API. Це дозволило категоризувати зібрані повідомлення за різними тональними ознаками (позитивна, негативна, нейтральна тощо). Розподіл тональностей не був рівномірним, що є типовим для реальних даних.

Зібраний і наданий для дослідження незбалансований тональний корпус текстів новин з Telegram-каналів, що містив 18270 постів у форматі json, є цінним ресурсом для аналізу настроїв та розробки методів автоматичного аналізу тональності. Використання інструментів Telegram API, Telethon, ElasticSearch, Kibana, а також приписування тональності за допомогою мовної моделі GPT-4-turbo забезпечує ефективність в роботі з великими обсягами текстової інформації. Цей корпус надає значні можливості для подальших досліджень та вдосконалення моделей аналізу тональності.

3.2. Опрацювання та аналіз попередньо зібраних даних (інформаційних повідомлень з Телеграм-каналів).

Після того, як був зібраний корпус текстів (18270 постів), його потрібно було підготувати до подальшого аналізу. Для виконання препроцесингу текстів було використано бібліотеку Stanza [52], розроблену Стенфордським університетом, яка відома своєю високою якістю обробки природної мови та підтримкою української мови. Процес препроцесингу включав токенізацію, лематизацію, визначення частин мови та видалення стоп-слів.

Насамперед було здійснено видалення порожніх повідомлень, а саме повідомлень, в яких мовна модель помилково провела очищення тексту. Після цього процесу ми отримали інформацію, що в очищеному датасеті міститься 16204 повідомлень.

	publish_date	source	text	named_entities	tonality
0	2024-03-07T14:09:09+00:00	Forbes Ukraine	Швеція офіційно стала 32-м членом НАТО. Відпов...	[Швеція, НАТО, Держдепартамент США]	позитивна
1	2024-03-19T16:57:17+00:00	Forbes Ukraine	«Державний оператор тилу» на початку лютого пр...	[«Державний оператор тилу», ЗСУ]	негативна
2	2024-03-07T11:21:30+00:00	Forbes Ukraine	142 людини відповіли, що готові підписатися, я...	[Forbes Ukraine]	позитивна
3	2024-03-19T16:42:14+00:00	Forbes Ukraine	Forbes шукає E-mail-маркетолога/-иню\n• запу...	[Forbes.ua, UNIT.City]	позитивна
4	2024-03-07T10:00:45+00:00	Forbes Ukraine	22 березня. Саміт експортерів: інструменти, но...	[Євген Осипов, Юрій Сорочинський, Юрій Риженко...	нейтральна

Малюнок 3.1. Приклад того, як були сформовані дані в датасеті (див. Додаток 1)

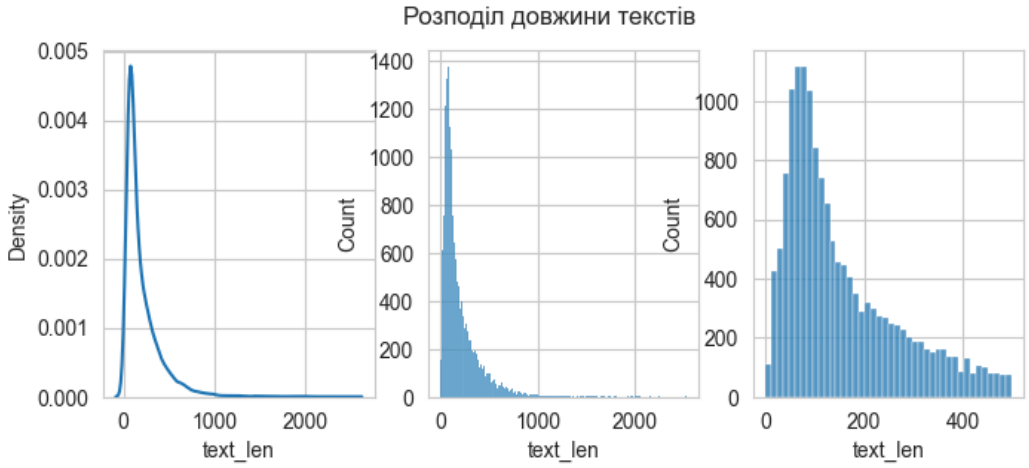
Важливими етапами у препроцесингу текстів є токенізація та лематизація, особливо для задач обробки природної мови (NLP). Токенізація – це розбиття тексту на менші об'єкти, які називаються токенами [34]. У лінгвістиці лематизація — це процес групування відмінюваних форм слова так, щоб їх можна було проаналізувати як єдиний елемент, ідентифікований за лемою слова або словниковою формою [33]. Це дозволяє зменшити різноманітність форм слів та спростити подальший аналіз текстів. Також для всіх текстів було здійснено визначення частин мови та видалення стоп-слів.

Так як ми мали доволі об’ємний корпус, то автоматична обробка текстів зайняла приблизно сім годин. Код, що використовувся для видалення стоп-слів, токенізації та лематизації наведений нижче.

```
# лематизація текстів
def lemmatize(text: str) -> List[Tuple[str, str]]:
    """
    Метод, що використовується для лематизації текстів
    :param text: вхідний текст
    :return: Tuple[str, str] - кортеж з леми та частини мови
    """
    nlp = Pipeline(lang='uk',
                   processors='tokenize,mwt,pos,lemma',
                download_method=stanza.DownloadMethod.REUSE_RESOURCES,
                   logging_level='error')
    doc = nlp(text)
    lemmas_pos = []
    for sent in doc.sentences:
        for word in sent.words:
            lemma = word.lemma
            pos = word.upos
            if (lemma.lower() not in stopwords) & (pos not in ['PUNCT', 'ADP', 'DET', 'INTJ',
'PRON', 'NUM']):
                lemmas_pos.append((lemma, pos))
    return lemmas_pos
# запуск перетворення усіх текстів
tqdm.pandas()
# якщо бінарний файл існує - прочитаємо його, якщо ні - створимо
binary_file = path_to_files / 'texts_lemmatized.pkl'
if binary_file.exists():
    print('Файл існує')
    df = pd.read_pickle(binary_file)
else:
    print('Файл буде створено')
    df['lemmas'] = df['text'].progress_apply(lemmatize)
    # запис текстів до бінарного файлу для подальшого використання
    df.to_pickle()
```
(див. Додаток 1)

Крім того, текстів з уже визначеною тональністю було 16148, це зумовлено тим, що мовна модель не завжди повертає результат, що пов'язано з налаштуваннями безпеки, а саме цензурою повідомлень та внутрішніми помилками API.

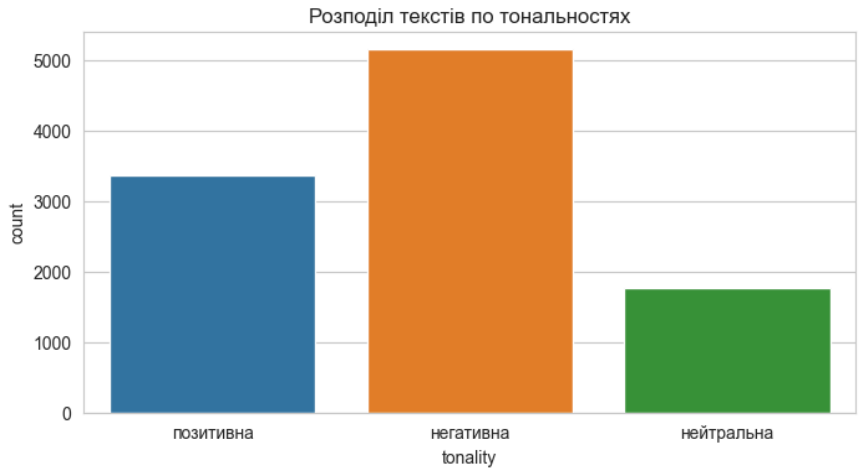
Також здійснювався аналіз розподілу довжини текстів, що є важливим кроком у дослідженні текстових даних, оскільки він дозволяє краще зрозуміти структуру та характеристику корпусу текстів. Таким чином було побудовано 3 графіки: щільність розподілу, гістограму всіх текстів та гістограму текстів з довжиною менше 500 символів. Отримуємо такі дані:



Малюнок 3.2. Розподіл довжини текстів (див. Додаток 1)

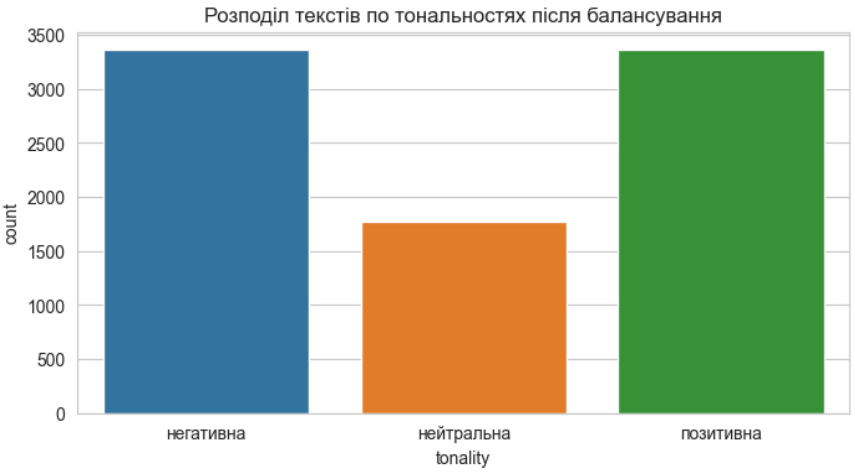
Можемо зробити такий висновок, що більшість текстів мають коротку довжину. Є кілька текстів середньої та великої довжини, але їх значно менше. Розподіл має різкий пік на початку (при коротких довжинах) і поступово зменшується для довших текстів.

На цьому етапі нам потрібно було обмежити кількість текстів за довжиною, оскільки тональність малих текстів складно визначити коректно, а дуже великі тексти, тобто довжина яких на порядок відрізняється від медіанної, будуть погіршувати навчання моделей. Тому було взято такі граничні точки — менше 80 та понад 500 символів. Це дозволяє зосередитися на текстах середньої довжини, які є більш придатними для аналізу тональності та моделювання. Таким чином ми отримали 10272 пости.



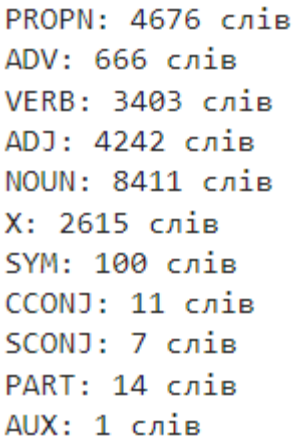
Малюнок 3.3. Розподіл текстів за тональністю (див. Додаток 1)

На цьому графіку ми можемо бачити незбалансованість класів, тому наступною нашою метою є їх збалансування. Потрібно зменшити кількість спостережень в найбільшому з них до рівня другого за кількістю класу. Після здійснення етапу ми отримуємо 8485 постів і таку характеристику даних:



Малюнок 3.4. Розподіл текстів за тональністю після збалансування вибірки (див. Додаток 1)

Останнім етапом, що був здійснений в обробці корпусу, став підрахунок унікальних слів загалом та за частинами мови. Отримуємо такі результати: кількість слів складає — 23707, а щодо кожної частини мови отримуємо наступні показники:



Малюнок 3.5. Результати підрахунку вживання кожної частини мови (див. Додаток 1)

Отже, препроцесинг текстових даних, що включав лематизацію, токенізацію, визначення частин мови та збалансування даних за тональністю, забезпечив такі переваги:

1) Зниження шуму в даних, а саме видалення малозначущих слів (стоп-слів) і приведення слів до їх базових форм.

2) Покращення якості моделювання — балансування класів зменшило вплив незбалансованості на результат майбутньої моделі.

3) Підвищення ефективності аналізу, так як токенізація та лематизація дозволять ефективніше аналізувати тексти.

Таким чином виконані кроки препроцесингу заклали надійну основу для подальшого аналізу тональності текстів та навчання моделей машинного навчання, що підвищить загальну точність і надійність результатів.

3.3. Визначення тональності текстів з використанням мовних моделей та моделей класифікаторів

У цьому підрозділі розглядаються методи визначення тональності текстів новин українською мовою за допомогою різних підходів та моделей класифікацій. Тональність тексту є ключовим елементом для розуміння емоційного забарвлення повідомлення, що має велике значення в аналізі

соціальних мереж, моніторингу громадської думки та інших застосунках обробки природної мови.

Основною метою є порівняння ефективності різних методів визначення тональності текстів українською мовою та виявлення найкращих підходів для конкретних завдань.

Першим підходом було обрано **визначення тональності з використанням тонального словника української мови.** Спочатку кожному слову з тексту присвоюють значення тональності зі словника (за умови його присутності в словнику), а потім обчислюють загальну тональність всього тексту шляхом підсумовування значення тональностей за кожною з емоційних категорій [20]. У дослідженні використовувався український тональний словник, розроблений групою осіб. Автори цього словника — Сергій Шеховцов, Олесь Петрів, Дмитро Чаплинський, Всеволод Дьомкін [46]. За цим словником ми отримували числовий показник тональності від 0 до 1, якщо тональність розраховано, або -1, якщо жодна з лем не знайдена в словнику.

У результаті ми отримали такі результати: тональність визначено для 3147 та невизначено для 5338 текстів, відповідне співвідношення можемо побачити на секторній діаграмі.

Малюнок 3.6. Співвідношення кейсів: визначено та не визначено (див. Додаток 2)

Підсумувавши, можемо стверджувати, що було використано тональний словник для тренування моделі визначення тональності текстів. Проте результати експерименту виявилися невтішними. Зокрема, для майже 63% текстів не було визначено тональність взагалі. Це може бути пов’язано з кількома факторами:

1) Недостатність тонального словника. Тональний словник, який використовувався для тренування, можливо, не охоплював весь спектр лексики та виразів, що використовуються у текстах. Це призвело до того, що значна частина текстів залишилася без визначеної тональності.

text	lemmas	tonality	tonality_score
У Білоцерківському районі проводяться роботи з...	[(білоцерківський, ADJ), (район, NOUN), (прово...	негативна	-1.000000
Та він же синючій, ледве стоїть на ногах! У Во...	[(синючий, ADJ), (стояти, VERB), (нога, NOUN),...	негативна	-1.000000
Затримки із західною допомогою в галузі безпек...	[(затримка, NOUN), (західний, ADJ), (допомога,...	негативна	0.806642
Скасовані авіарейси і тисячі людей без світла:...	[(скасований, ADJ), (авіареїс, NOUN), (світло,...	негативна	-1.000000
Посольство США в Росії закликало громадян уник...	[(посольство, NOUN), (США, PROPN), (Росія, PRO...	негативна	-1.000000

Малюнок 3.7. Розраховані оцінки тональності за допомогою тонального словника української мови (див. Додаток 2)

Як бачимо з наведених прикладів, для чотирьох текстів з п’яти модель не знайшла жодної з переліку леми у тональному словнику.

2) Складність природної мови, тобто тексти можуть містити складні мовні конструкції, сарказм, іронію або контекстуальні нюанси, які важко обробляти за допомогою простого словника. Тональний словник може не враховувати контекст, що важливо для коректного визначення тональності.

Щодо текстів, яким була приписана тональність, лише 50,5 % з них були визначені правильно. Таким чином, загальний відсоток правильно визначеної тональності склав лише 19 % від загальної кількості текстів, що є неприпустимо низьким результатом. Це вказує на те, що поточна модель потребує значного вдосконалення для досягнення прийнятного рівня точності.

Інший підхід, що використовувався у дослідженні — **визначення тональності за ембендингами слів, що згенеровані локально розгорнутою мовною моделлю BERT** [36] (англ. Bidirectional Encoder Representations from

Transformers). **BERT** — це сучасна модель глибокого навчання, розроблена Google, яка досягла значного прогресу в обробці природної мови (NLP).

Першим кроком в аналізі тональності текстів було використання токенізатора BERT, який розбивав текст на окремі токени, які могли бути словами або частинами слів. Це дозволяло моделі BERT ефективно обробляти текст, враховуючи контекст кожного токена в реченні.

Наступним кроком було створення масок уваги (attention masks). Маска уваги — це бінарна маска, яка визначає, на які токени слід звернути увагу (з ненульовою вагою), а які слід проігнорувати (з нульовою вагою) [38]. Це дозволяло моделі зосереджувати увагу на важливих частинах тексту та ігнорувати непотрібні елементи, що покращує якість навчання та передбачення.

Після токенізації та створення масок уваги, модель BERT була навчена з використанням бібліотеки PyTorch [53], що надає потужні інструменти для роботи з нейронними мережами та глибоким навчанням, дозволяючи ефективно навчати модель на великих обсягах даних. Під час навчання модель адаптується до специфічних особливостей тональності текстів, що дозволяє покращити точність передбачень.

tokenized_text	bert_emb
[спікер, палата, представник, США, Майк, Джонс...	[0.042942617, 0.15124951, -0.27168947, 0.11622...
[український, енергетика, лічений, година, від...	[0.22348423, 0.17753334, -0.05827129, 0.329789...
[чемпіонат, Європа, зіграти, група, суперник, ...	[0.05663139, 0.18377115, -0.25358963, 0.167722...

Малюнок 3.8. Отримані токени та ембедінги (див. Додаток 3)

На основі отриманих вихідних представлень текстів за допомогою BERT, був навчений класифікатор XGBoost (англ. Extreme Gradient Boosting) [62] — це інструмент для задач класифікації, який побудований на методі градієнтного

бустингу, що полягає в поступовій побудові ансамблю слабких моделей, зазвичай дерев рішень, де кожне наступне дерево намагається виправити помилки попередніх дерев. Отже, кінцева модель, яка є сукупністю всіх цих дерев, може робити більш точні прогнози, ніж будь-яке окреме дерево рішень. Використання XGBoost дозволило додатково покращити точність моделі, комбінуючи силу глибокого навчання та ефективності бустингу.

Насамперед ми кодували текстові мітки в числові, формували матрицю ознак (вектори BERT-представлень, які зберігалися в полі “bert_emb”) та масив міток, а потім розділяли дані на тренувальні та тестові набори з використанням train_test_split, де 20 % даних виділося для тестування:

```
df['label'] = LabelEncoder().fit_transform(df['tonality'].values)
X, y = np.vstack(df['bert_emb'].values), df['label'].values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

(див. Додаток 3)

Потім створювали об'єкти DMatrix для тренувальних та тестових даних та встановлювали наступні параметри для моделі XGBoost:

- objective: *multi:softmax* вказує, що задача є багатокласовою класифікацією з передбаченням одного класу.
- num_class: *len(set(y_train))* вказує кількість класів у мітках.
- eta: *0.01*, що задає швидкість навчання (англ. learning rate).
- max_depth: *10* встановлює максимальну глибину дерев рішень.

Процес навчання моделі на тренувальних даних запускався протягом 200 ітерацій, після чого отримали передбачені мітки для тестових даних [2. 0. 0. ... 2. 1. 0.].

Останнім етапом було отримати повний звіт про класифікацію моделі XGBoost на тестових даних. Ми використовували функцію classification_report з бібліотеки scikit-learn [56], щоб обчислити основні метрики: точність (англ. precision), повноту (англ. recall), F1-міру (англ. F1 score) та підтримку (англ. support) для кожного класу.

```
report = classification_report(y_test, preds, digits=3)
print(report)
```

	precision	recall	f1-score	support
0	0.513	0.594	0.551	697
1	0.229	0.046	0.076	351
2	0.477	0.602	0.532	649
accuracy			0.484	1697
macro avg	0.406	0.414	0.386	1697
weighted avg	0.440	0.484	0.445	1697

Малюнок 3.9. Результати натренованої моделі (див. Додаток 3)

З отриманих результатів можемо зробити такі висновки: модель має проблеми з класифікацією класу 1 (мітка нейтрального тексту), оскільки значення точності, повноти та F1-Score для цього класу є дуже низькими. Це може бути через недостатню кількість даних для цього класу або через особливості даних, які роблять класифікацію складною. Класи 0 (мітка негативного тексту) і 2 (мітка позитивного тексту) класифікуються значно краще, але все одно є місце для покращення.

Загальна точність (англ. accuracy) моделі становить 48,4%, що свідчить про те, що модель правильно класифікує майже половину всіх прикладів. Макро середнє (англ. macro-averaging) для всіх класів без урахування їх кількості відображає низькі значення, що вказують на погану продуктивність в середньому для всіх класів. Зважене середнє (англ. weighted averaging) враховує кількість кожного класу у вибірці. Показники трохи кращі за середнє значення, але все ще невисокі.

Отримані показники значно перевищують попередні результати, однак все ще недостатні для якісної роботи у подальших цілях. Вдосконалення можуть включати додаткове налаштування параметрів моделі, використання більшого обсягу даних для навчання або застосування інших методів покращення якості аналізу тональності.

У наступному дослідженні використано підхід, що включає **визначення тональності за допомогою попередньо натренованої (fine-tuning) моделі Gemini-1.0-pro** [49]. Fine-tuning попередньо натренованої моделі за допомогою

Gemini API дозволяє адаптувати великі мовні моделі (LLMs) до конкретних завдань або доменів. Це процес, за допомогою якого вже навчену модель додатково тренують на специфічних даних, щоб поліпшити її продуктивність в певному контексті.

Спершу було завантажено облікові дані з файлу «service_secret.json» для подальшої аутентифікації під час викликів API і налаштовано бібліотеку genai [44]. Потім із наявних моделей було вибрано базову модель, яка підтримує налаштування.

```
credentials = service_account.Credentials.from_service_account_file('./service_secret.json')
genai.configure(credentials=credentials)
base_model = [
    m for m in genai.list_models()
    if "createTunedModel" in m.supported_generation_methods][0]
base_model
```
(див. Додаток 4)

Вхідні дані було розділено на навчальний і тестовий набори за допомогою функції train_test_split, де на тренувальну частину було виділено 95% і на тестову 5% вибірки, після чого сформовано навчальний набір даних у вигляді списку словників, де кожен словник містить пару “text_inpu”" (текстовий запит) і “output” (відповідь).

```
X, y = df['prompt'], df['response']
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.95, random_state=42)
train_data = []
for text_input, output in zip(X_train, y_train):
    train_data.append({'text_input': text_input, 'output': output})
```
(див. Додаток 4)

Далі виконувалася спроба завантажити налаштовану модель «sentiment-analysis-upd», а в разі невдачі було створено нову модель із заданими параметрами навчання, до якої створили екземпляр генеративної AI-моделі, налаштованої для аналізу тональності.

Використовували метод create_tuned_model з наступними параметрами:

- source_model = base_model.name: Базова модель, на основі якої буде створено настроєну модель.
- training_data = train_data: Тренувальні дані у вигляді списку словників.
- id = NAME: Ідентифікатор нової моделі.

- epoch_count = 5: Кількість епох тренування.
- batch_size = 32: Розмір батча під час тренування.
- learning_rate = 0.0002: Швидкість навчання.

Також задіювали метод wait_bar для відстеження стану операції створення моделі. У кожній ітерації циклу for виконується пауза тривалістю 30 секунд перед наступною перевіркою.

```
NAME = 'sentiment-analysis-upd'
try:
  model = genai.get_tuned_model(f'tunedModels/sentiment-analysis-upd')
  if model.state != 2:
    raise RuntimeError('The model can not be used')
  else:
    print('Модель можна використовувати')
except Exception as ex:
  print(ex)
  operation = genai.create_tuned_model(
    source_model=base_model.name,
    training_data=train_data,
    id = NAME,
    epoch_count = 5,
    batch_size=32,
    learning_rate=0.0002,
  )
  for status in operation.wait_bar():
    time.sleep(30)
```

(див. Додаток 4)

Після налаштування модель було використано для аналізу на тестовому наборі даних. Отримані відповіді моделі зберігалися разом із вхідними даними і очікуваними результатами у список «answers_scope», який потім було перетворено у DataFrame і збережено у файл у форматі *pickle*.

```
answers_scope = []
counter = 0
for input_text, output in tqdm(zip(X_test, y_test), total=len(X_test), desc='Tuned model test'):
  if (counter % 14 == 0) & (counter != 0):
    time.sleep(60)
  response = answer_model.generate_content(input_text)
  try:
      answers_scope.append({'input': input_text, 'expected_output': output, 'model_output': response.text})
  except Exception as ex:
```

print(ex)
answers_scope.append({'input': input_text, 'expected_output': output, 'model_output': None}) (див. Додаток 4)

	input	expected_output	model_output
0	Визнач тональність тексту, користуючись тернар...	нейтральна	позитивна
1	Визнач тональність тексту, користуючись тернар...	негативна	негативна
2	Визнач тональність тексту, користуючись тернар...	негативна	негативна
3	Визнач тональність тексту, користуючись тернар...	позитивна	позитивна
4	Визнач тональність тексту, користуючись тернар...	позитивна	позитивна

Малюнок 3.10. Отримані значення генеративної моделі для визначення тональності

Наступним етапом було виконати оцінку точності налаштованої моделі шляхом порівняння її прогнозованих відповідей з реальними даними. Кожен запис у тестовому наборі даних перевіряється на збіг між очікуваним виводом (відповідь з тестового набору) та виводом, який згенерувала модель. Якщо ці значення співпадають, значення 'label' для цього запису буде True, інакше — False. Після створення стовпця 'label' обчислюється кількість входжень True та False у цьому стовпці. Це дозволяє нам зрозуміти, скільки разів модель правильно передбачила результат, а скільки разів помилилася. За допомогою підрахунків відношення правильних відповідей до загальної кількості випадків визначається точність моделі. Ця точність виводиться у вигляді десяткового числа, що описує, яка частка відповідей моделі є правильними.

Окрім точності, виводиться також підрахунок кількості True та False. Це дає можливість докладніше оцінити, наскільки часто модель правильно передбачає результати. Отримали такі результати : точність (англ. Accuracy) — 0.772, правильні результати — 325 та неправильні — 96.

df['label'] = df_test[['expected_output', 'model_output']].apply(lambda x: x.iloc[0]==x.iloc[1], axis=1)
count = df['label'].value_counts()

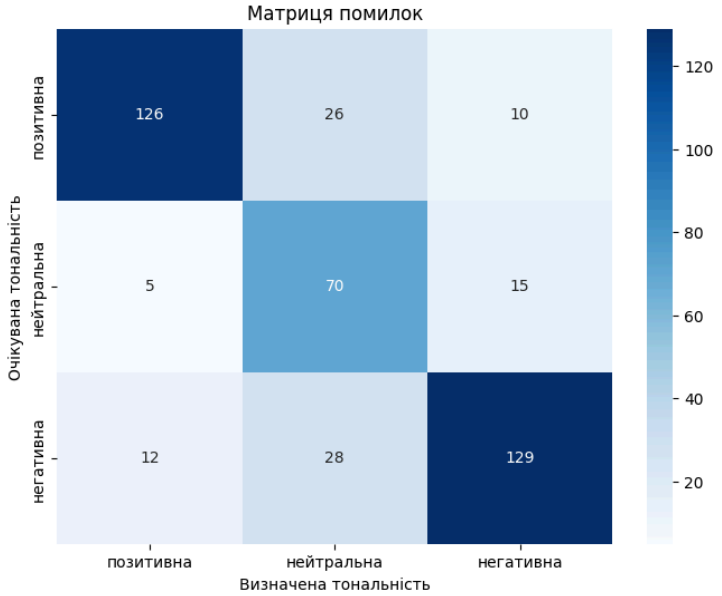
```
print(f'Accuracy = {(count[True] / (count.sum())):.3f}')
count
```

Результат після запуску:

```
Accuracy = 0.772
       label
True     325
False     96
Name: count, dtype: int64
```

(див. Додаток 4)

Останнім етапом було створення візуалізації матриці помилок для моделі класифікації тональності тексту. Матриця помилок допомагає оцінити продуктивність моделі, показуючи кількість правильних і неправильних класифікацій для кожної категорії.



Малюнок 3.11. Матриця помилок (див. Додаток 4)

Можемо зробити кілька висновків.

Модель досить добре класифікує позитивні тексти, з точністю 126/162 (близько 77,8 %). Однак, значна кількість позитивних текстів помилково класифікується як нейтральні (26 випадків) і меншою мірою як негативні (10 випадків).

Модель менш ефективно визначає нейтральні тексти, з точністю 70/90 (близько 77,8 %). Вона плутає нейтральні тексти як з позитивними (5 випадків), так і з негативними (15 випадків).

Модель найбільш ефективно класифікує негативні тексти, з точністю 129/169 (близько 76,3 %). Проте деяка кількість негативних текстів класифікується як нейтральні (28 випадків) і позитивні (12 випадків).

Загалом, модель показує набагто кращі результати, аніж попередні два експерименти, але є простір для покращення, особливо у класифікації нейтральних текстів.

Наш сервісний акаунт для використання Google API був заблокований з причин порушення умов використання та підозрілої активності. Тому цю модель ми не зможемо використовувати.

В останньому експерименті **використовувалась кастомна модель з платформи HuggingFace [50], адаптована під українську мову. Це модель, що заснована на архітектурі RoBERTa** (англ. Robustly optimized BERT approach) [55], яка була спеціально підготовлена для обробки української мови з метою визначення частин мови (англ. UPOS — Universal Part-of-Speech). Архітектура RoBERTa — це вдосконалена архітектура BERT з покращеною тренувальною стратегією, яка включає в себе більше даних та довший час тренування.

Процес роботи передбачав кілька етапів.

Підготовка даних до роботи, а саме файл gemini_full.csv, який ми використовували для навчання. Ми перейменували стовпці для узгодженості з форматами Hugging Face, замінили текстові мітки класів (негативна, нейтральна, позитивна) на числові (0, 1, 2). Також розподіли дані на тренувальну та тестову вибірки та перетворили їх у формат Dataset бібліотеки datasets [37]:

```
#Завантаження даних
file_path = 'gemini_full.csv'
data = pd.read_csv(file_path)
#Переведення у формат, адаптований для Hugging Face
data = data.rename(columns={"prompt": "text", "response": "label"})
label_mapping = {'негативна': 0, 'нейтральна': 1, 'позитивна': 2}
data['label'] = data['label'].map(label_mapping)
#Тренувальна/тестова вибірка
train_data, test_data = train_test_split(data, test_size=0.2, random_state=42)
```

```
#Об'єкти Dataset
train_dataset = Dataset.from_pandas(train_data)
test_dataset = Dataset.from_pandas(test_data)
```
(див. Додаток 5)

Далі ми здійснювали ініціалізацію токенізатора та моделі roberta-base-ukrainian від KoichiYasuoka та встановлювали кількість вихідних класів для моделі, а саме 3 класи для класифікації тональності: негативна, нейтральна, позитивна. Застосовували функцію токенізації до тренувальної і тестової вибірок, після чого видаляли зайві стовпці, які не потрібні для навчання (текст і індекс) і отримали дані у форматі, який підтримує PyTorch.

```
#Токенізатор та модель для класифікації
model_name = "KoichiYasuoka/roberta-base-ukrainian" #назва моделі, база: roberta-base
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForSequenceClassification.from_pretrained(model_name, num_labels=3)
#Токенізація даних
def tokenize_function(examples):
    return tokenizer(examples['text'], padding="max_length", truncation=True)
train_dataset = train_dataset.map(tokenize_function, batched=True)
test_dataset = test_dataset.map(tokenize_function, batched=True)
#Видалення зайвих стовпців
train_dataset = train_dataset.remove_columns(["text", "__index_level_0__"])
test_dataset = test_dataset.remove_columns(["text", "__index_level_0__"])
train_dataset.set_format("torch")
test_dataset.set_format("torch")
```
(див. Додаток 5)

Наступним етапом було виконати налаштування та тренування моделі для класифікації текстів. Перш за все було обрано метрики для моніторингу продуктивності моделі під час тренування. Використовували функцію compute_metrics, що визначала точність та F1-мітку.

```
from sklearn.metrics import accuracy_score, f1_score
#Метрики для моніторингу якості
def compute_metrics(p):
    preds = p.predictions.argmax(-1)
    accuracy = accuracy_score(p.label_ids, preds)
    f1 = f1_score(p.label_ids, preds, average='weighted')
    return {"accuracy": accuracy, "f1": f1}
```
(див. Додаток 5)

Для налаштування моделі були обрані такі параметри:

- evaluation_strategy — стратегія оцінювання моделі була обрана після кожної епохи.

- learning_rate — швидкість навчання, 2e-5
- num_train_epochs — кількість епох навчання 5.
- weight_decay — параметр для регуляризації (запобігання перенавчанню) 0.01.
- warmup_steps — кількість кроків для розігріву, щоб модель поступово адаптувалася до навчання — 500

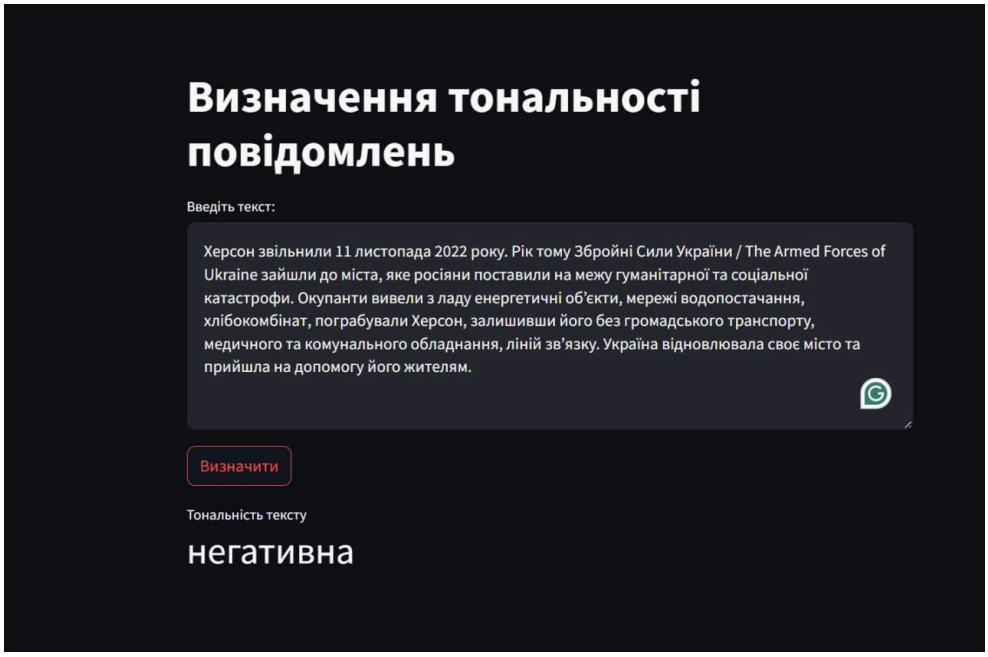
Після тренування результати були такі:

Номер епохи навчання	Точність	Мітка F1
1	0.594	0.590
2	0.643	0.623
3	0.646	0.632
4	0.642	0.641
5	0.642	0.636

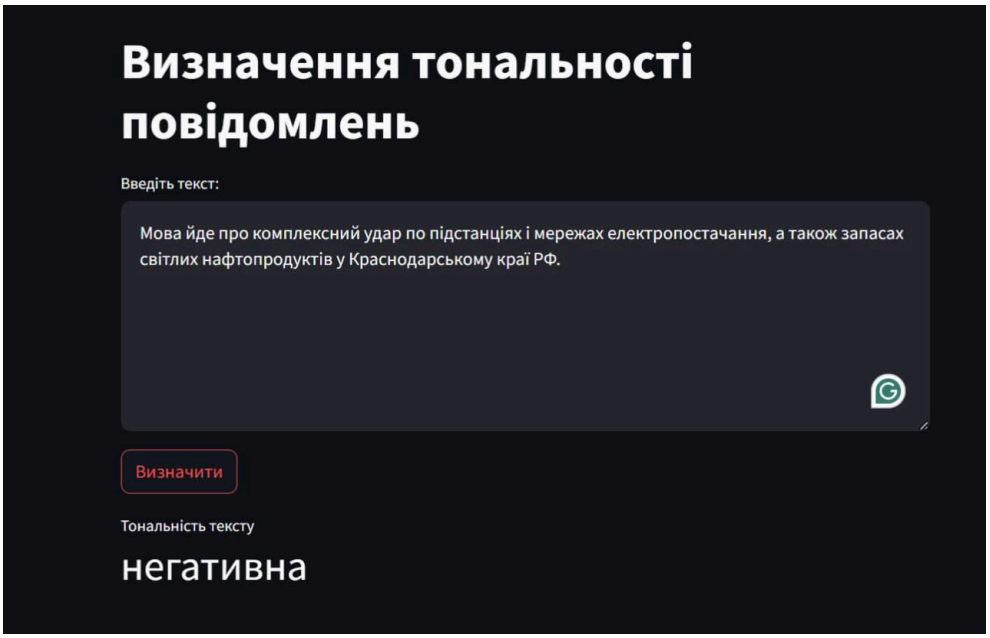
Малюнок 3.12. Оцінка роботи моделі на кожній епосі навчання

Згідно з таблицею, модель продемонструвала позитивну динаміку показників точності та F1-мітки з кожною епохою навчання. Це свідчить про те, що модель ефективно навчалася і покращувала свої прогнози. Точність моделі зросла з 0.594 до 0.642 протягом 5 епох навчання. Це означає, що модель стала краще класифікувати дані з часом. F1-мітка моделі зросла з 0,590 до 0,641 протягом 5 епох навчання.

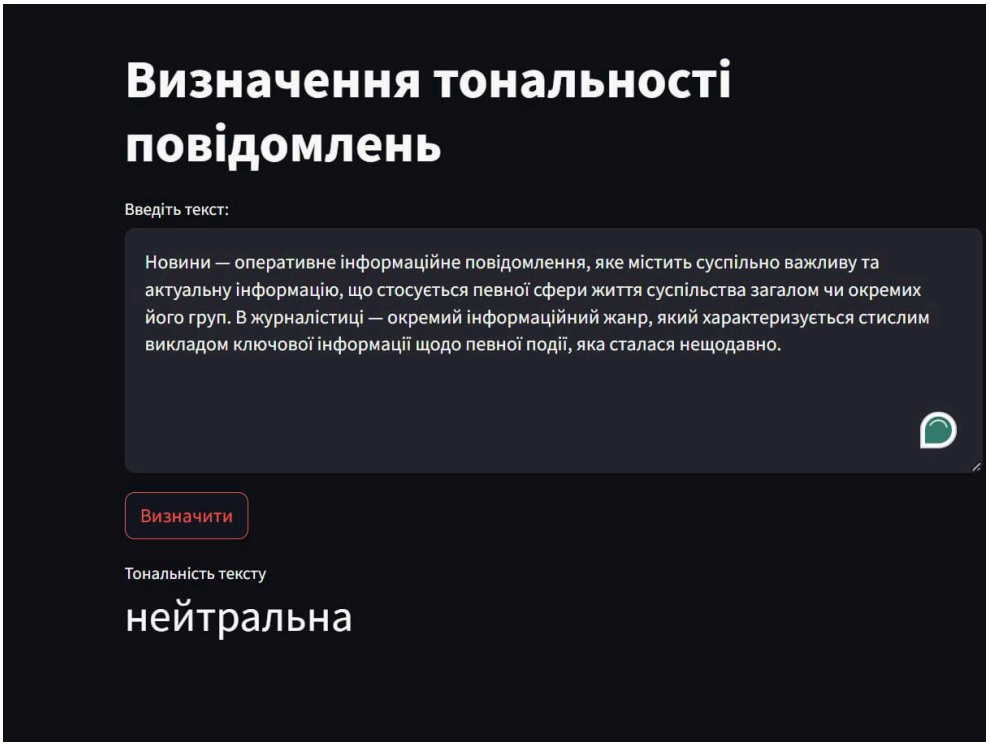
Для тестування нашої моделі було створено вебзастосунок (див. Додаток 5) для визначення тональності за допомогою бібліотеки Streamlit [60], у якому користувач може ввести текст у текстове поле і натиснути кнопку «Визначити». Застосунок використовує модель для аналізу введеного тексту і визначення його тональності (позитивна, нейтральна або негативна). Результат відображається на екрані. Приклади правильної класифікації:



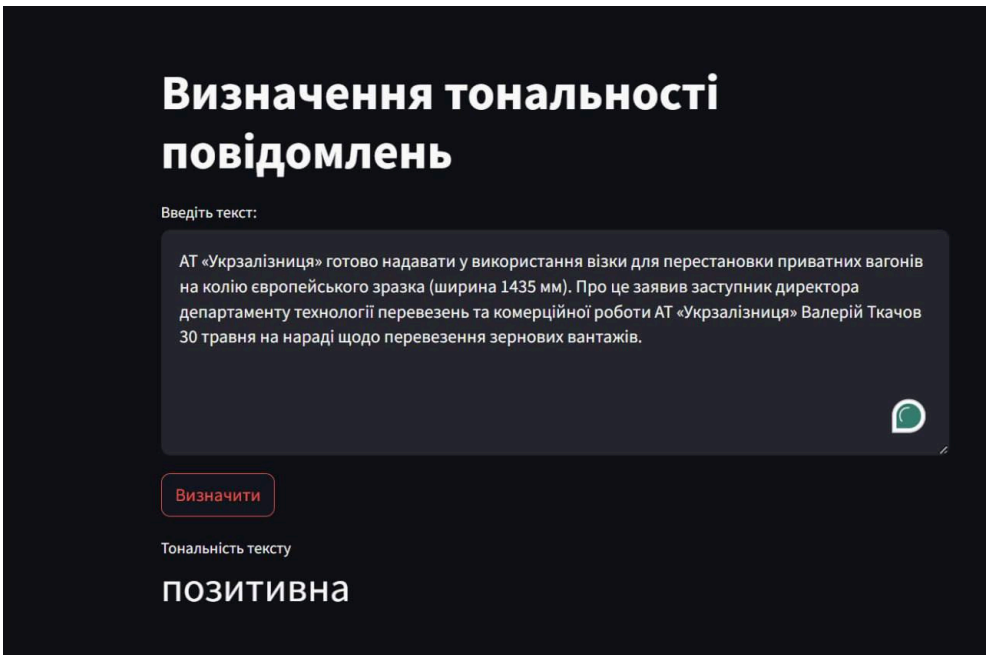
Малюнок 3.13. Приклад роботи застосунку №1



Малюнок 3.14. Приклад роботи застосунку №2

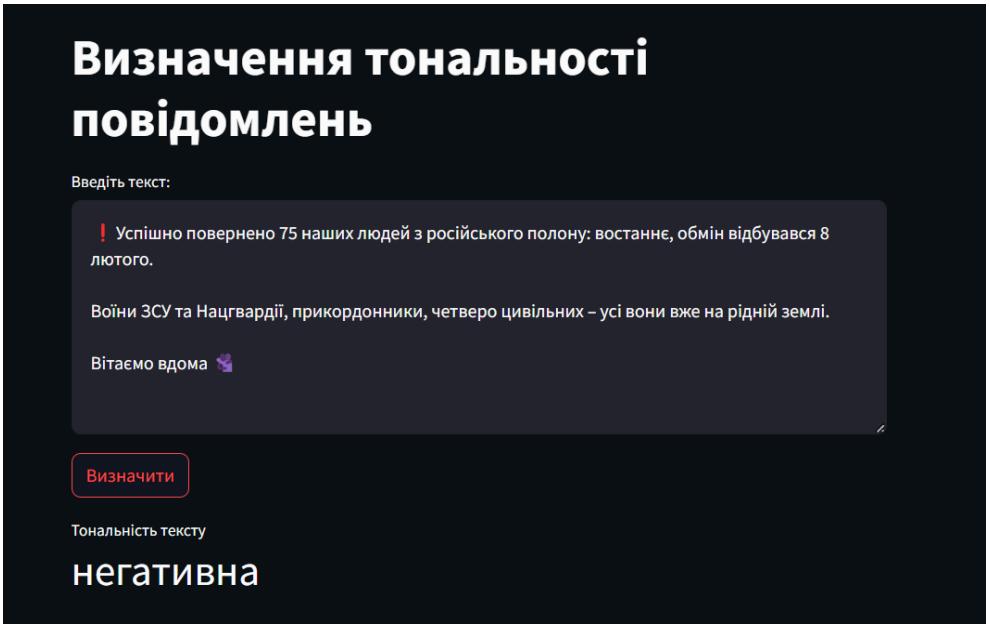


Малюнок 3.15. Приклад роботи застосунку №3



Малюнок 3.16. Приклад роботи застосунку №4

Модель може помилятись, оскільки не має високої точності. Наприклад, у таких контекстах.



Малюнок 3.17. Приклад роботи застосунку №5

Для покращення результатів моделі визначення тональності можна використати наступні підходи: додавання більшого обсягу різноманітних даних, що дозволить моделі краще розрізняти варіанти виразу емоцій та контексту. Використання сучасних моделей, таких як BERT, Gemini або GPT, забезпечить краще узагальнення та здатність розуміти смислові зв'язки. Оптимізація параметрів, таких як швидкість навчання, розмір батча та розмір шарів, зможе покращити точність моделі. А також об'єднання кількох моделей для отримання узагальнених та більш точних результатів.

Висновки до розділу 3

У результаті проведеного дослідження ми отримали декілька важливих висновків, що розширюють наше розуміння процесу обробки текстів новин з Telegram і визначення їх тональності.

По-перше, ми з'ясували, що обробка текстів займає значну кількість часу, особливо коли мова йде про великі обсяги даних. Етапи підготовки, очищення, попередньої обробки текстів та збалансування датасету вимагають ретельного підходу, щоб забезпечити високу якість вхідних даних для подальшого аналізу.

По-друге, ми дослідили чотири різні підходи до автоматизованого визначення тональності текстів: тональний словник, визначення тональності за

ембендингами слів, що згенеровані локально розгорнутою мовною моделлю BERT, за допомогою попередньо натренованої моделі (fine-tuning) Gemini-1.0-pro та за допомогою попередньо натренованої моделі (fine-tuning) RoBERTa.

Результати дослідження показали, що найкращі результати були досягнуті за допомогою моделі Gemini-1.0-pro та за допомогою кастомної моделі з платформи HuggingFace, на основі архітектури RoBERTa. Це свідчить про значний прогрес у сфері машинного навчання і підкреслює ефективність сучасних передових моделей для завдань аналізу тональності. Використання попередньо натренованих моделей дозволяє досягти високої точності та надійності в порівнянні з більш традиційними методами.

На основі отриманих результатів останньої моделі було створено застосунок для автоматизованого визначення тональності новин з Telegram. Цей інструмент може бути корисним для різних категорій користувачів: журналістів та редакторів, аналітиків та дослідників, політиків та їхніх команд, які можуть використовувати його для швидкого аналізу тональності новинних текстів і відслідковування громадської думки.

У процесі проведення цього дослідження було розглянуто широкий спектр літератури та онлайн ресурсів, а саме документація та ресурси по роботі зі Stanza, великі мовні моделі та їх документація, онлайн спільноти та форуми, такі як Stack Overflow [59], GitHub [40] та Reddit [54], а також наукові публікації, які надавали сучасні дослідження та розробки у сфері NLP та машинного навчання [1, 7, 8, 11, 51]. Ці ресурси забезпечили не лише теоретичні знання, але й практичні навички, що були критично важливими для успішної реалізації проекту.

Висновки

У процесі виконання бакалаврської роботи було здійснено комплексний аналіз та дослідження автоматичного тонального аналізу текстів, зокрема, визначення тональності текстів новин з Telegram-каналів. У трьох основних розділах було розглянуто ключові аспекти цієї галузі, методи аналізу та результати застосування різних підходів.

У результаті ґрунтовного огляду літератури було поглиблено розуміння галузі аналізу тональності текстів — основні поняття, завдання та проблематика. Ознайомлення з ключовими працями, такими як "Sentiment Analysis and Opinion Mining" Б. Лю, "Challenges in Sentiment Analysis" Саїфа М. Мохаммада та "Opinion mining and sentiment analysis" Бо Панга й Ліліан Лі, надало цінні знання у цій сфері. Огляд літератури сформував міцну теоретичну базу для подальшої практичної реалізації.

Вивчення різноманітних методів та алгоритмів аналізу тональності текстів, включно з традиційними підходами на основі словників та новітніми методами машинного навчання, такими як нейронні мережі та великі мовні моделі, дозволило визначити найбільш перспективні напрямки для подальшої роботи. На основі аналізу було обрано чотири конкретні підходи: класичний метод із використанням тонального словника, модель BERT для генерації ембедингів слів, попередньо натреновану велику мовну модель Gemini-1.0-pro та підхід на основі глибокого навчання для аналізу тональності з використанням попередньо навченої трансформерної моделі.

У ході проведеного дослідження було реалізовано та протестовано обрані методи на наборі текстових новин з Telegram. Результати експериментів засвідчили, що найвищу точність продемонструвала модель Gemini-1.0-pro — 0.772, що підтверджує ефективність сучасних передових моделей машинного навчання для завдань аналізу тональності, проте ми не встигли її достатньо протестувати. Також хороший результат показало навчання з використанням трансформера RoBERTa, а саме точність від 0.593 до 0.645. На основі цього

підходу було створено застосунок для автоматизованого визначення тональності новин.

Теоретичні аспекти та методологічні підходи, розглянуті у роботі, надають глибоке розуміння проблем та можливостей цієї галузі, вказуючи на перспективні напрямки подальших досліджень і розвитку технологій. Розроблений застосунок демонструє значний потенціал для широкого застосування автоматичного тонального аналізу в реальних умовах, забезпечуючи зручність і точність обробки текстової інформації.

Список використаних джерел

Джерела

1. Метод інтелектуального аналізу емоційної тональності текстової інформації для визначення поведінкових намірів нейромережевими засобами И / О. Залуцька та ін. *Вісник Хмельницького національного університету.* 2023. Т. 1, № 5. С. 67–73. URL: https://elar.khmnu.edu.ua/server/api/core/bitstreams/7d4a9f0d-53d5-4633-bf69-e360e5206a70/content.
2. Рябишев О., Єрохін А., Бахмет А. Аналіз тональності тексту українською мовою. *Бионика Интеллекта.* 2021. №1 С. 15–21. URL: https://openarchive.nure.ua/server/api/core/bitstreams/f59561cb-66a5-4f99-b6ff-cf30a5162f91/content.
3. Система електронного забезпечення навчання ЗНУ. URL: https://moodle.znu.edu.ua/pluginfile.php?file=/486136/mod_resource/content/1/Лекція%209.pdf.
4. Сіріньок-Долгарьова К. Г. Російська дезінформація і пропаганда в соціальних медіа: кейс Телеграму. Протидія дезінформації в умовах російської агресії проти України: виклики і перспективи. *Науково-дослідний інститут публічної політики і соціальних наук.* 2023. С. 153-156 URL: https://doi.org/10.32782/ppss.2023.1.40
5. Bhonde S. B., Prasad J. R. Sentiment Analysis - Methods, Applications & Challenges. *International Journal of Electronics Communication and Computer Engineering*. Vol. 6, Issue 6. P. 634–640. URL: https://ijecce.org/administrator/components/com_jresearch/files/publications/IJECCE_3633_Final.pdf
6. Boiy E., Moens M.-F. A machine learning approach to sentiment analysis in multilingual Web texts. *Information Retrieval.* 2008. Vol. 12, no. 5. P. 526–558. URL: https://doi.org/10.1007/s10791-008-9070-z.

7. Buscemi A., Proverbio D. ChatGPT vs Gemini vs LLaMA on Multilingual Sentiment Analysis. *Department of Industrial Engineering, University of Trento*. P. 1–11. URL: https://arxiv.org/pdf/2402.01715.
8. Islam R., Ahmed I. Gemini-the most powerful LLM: Myth or Trut. Dept of *Computer Science & Engineering*. P. 1–7. URL: https://www.techrxiv.org/doi/full/10.36227/techrxiv.171177477.70151414.
9. Jasper H., Berrick Eldridge K. Automated Social Media Research and Sentiment Analysis using LLMs. *The University of Hong Kong Department of Computer Science*. 2024. P. 1–28. URL: https://wp2023.cs.hku.hk/fyp23047/wp-content/uploads/sites/48/fyp23047-interim-report.pdf.
10. Learning Subjective Language / J. Wiebe et al. Computational Linguistics. 2004. Vol. 30, no. 3. P. 277–308. URL: https://doi.org/10.1162/0891201041850885.
11. Liladhar Rane N., Choudhary S. P., Rane J. Gemini or ChatGPT? Capability, Performance, and Selection of Cutting-edge Generative Artificial Intelligence (AI) in Business Management. *Studies in Economics and Business Relations*. 2024. Vol. 5, no. 1. P. 40–50. URL: https://deliverypdf.ssrn.com/delivery.php?ID=851091091009085080093120114103079099034023058067019062072065006100008101081023100123034016097101060099003108124099012092118081026058012038004031001115109105075003101042026047065079123072094094088070012126099103124078010092027023084025018075123100085069&EXT=pdf&INDEX=TRUE.
12. Liu B. Sentiment Analysis and Opinion Mining. Cham : *Morgan & Claypool Publishers,* 2012. 168c. URL: https://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf.
13. Medhat W., Hassan A., Korashy H. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*. 2014. Vol. 5, no. 4. P. 1093–1113. URL: https://doi.org/10.1016/j.asej.2014.04.011.

14. Mohammad S. M. Challenges in Sentiment Analysis. *A Practical Guide to Sentiment Analysis*. Cham, 2017. P. 61–83. URL: https://doi.org/10.1007/978-3-319-55394-8_4.
15. Pang B., Lee L. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*. 2008. Vol. 2,, No 1-2. P. 1–135. URL: https://www.cs.cornell.edu/home/llee/omsa/omsa.pdf.
16. S. Kusal et al. A Review on Text-Based Emotion Detection - Techniques, Applications, Datasets, and Future Directions. 2022. URL: https://arxiv.org/pdf/1904.02232.
17. Saad S., Saberi B. Sentiment Analysis or Opinion Mining: A Review. *International Journal on Advanced Science Engineering and Information Technology*. 2017. Vol. 7, No 5. P. 1660–1666. URL: https://www.researchgate.net/publication/320762707_Sentiment_Analysis_or_Opinion_Mining_A_Review.
18. Sadia A., Khan F., Bashir F. An Overview of Lexicon-Based Approach For Sentiment Analysi. *International Electrical Engineering Conference* Karachi, 3 February 2024. URL: https://ieec.neduet.edu.pk/2018/Papers_2018/15.pdf.
19. Sharma A., Dey S. Performance Investigation of Feature Selection Methods and Sentiment Lexicons for Sentiment Analysis. *Special Issue of International Journal of Computer Applications*. 2012. P. 15–20. URL: https://www.researchgate.net/profile/Shubhamoy-Dey/publication/302280130_Performance_Investigation_of_Feature_Selection_Methods_and_Sentiment_Lexicons_for_Sentiment_Analysis/links/572f4aa708ae3736095c1876/Performance-Investigation-of-Feature-Selection-Methods-and-Sentiment-Lexicons-for-Sentiment-Analysis.pdf.
20. System of automatic determination of text tone / I. Olenych et al. *Electronics and Information Technologies*. 2021. Vol. 15. URL: https://doi.org/10.30970/eli.15.2

21. Wankhade M., Rao A. C. S., Kulkarni C. A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review*. 2022. URL: https://link.springer.com/article/10.1007/s10462-022-10144-1.
22. Wilfred Kiprono K. Comparative Twitter Sentiment Analysis Based on Linear and Probabilistic Models. *International Journal on Data Science and Technology*. 2016. Vol. 2, no. 4. P. 41. URL: https://doi.org/10.11648/j.ijdst.20160204.11.

Інтернет-ресурси

23. Ароматична веганська свічка "Mandarine Punch" - MAREVE: купити за найкращою ціною в Україні | *Makeup.ua.*. URL: https://makeup.com.ua/ua/product/1057316/.
24. Бізнес України. *Telegram*. URL: https://t.me/businessua.
25. Загальна інформація про Elastic. *Хостинг Україна*. URL: https://www.ukraine.com.ua/uk/wiki/elastic/overview/.
26. Київ Оперативний | Kyiv Operative. *Telegram*. URL: https://t.me/kyivoperat.
27. Нейромережа – що це таке, як працює та навіщо потрібна. *Termin.in.ua*. URL: https://termin.in.ua/neyromerezha/.
28. Новинарня. *Telegram*. URL: https://t.me/Novynarnia.
29. Реальний Київ | Украина. *Telegram*. URL: https://t.me/kievreal1.
30. Спорт України. *Telegram*. URL: https://t.me/sportukraineUA.
31. Труха Україна. *Telegram*. URL: https://t.me/truexanewsua.
32. ТСН новини / ТСН.ua. *Telegram*. URL: https://t.me/TCH_channel.
33. Учасники проектів Вікімедіа. Лематизація – Вікіпедія. *Вікіпедія*. URL: https://uk.wikipedia.org/wiki/Лематизація.
34. API токенізації та лематизації на основі spaCy. *Advanced Artificial Intelligence API*. URL: https://nlpcloud.com/uk/nlp-tokenization-api.html.
35. Avramenko D. Теорема Баєса: виведення і розуміння. *Medium*. URL: https://medium.com/@dmytro.avramenko/теорема-баєса-виведення-і-розуміння-599cec628e96.

36. BERT. *Hugging Face – The AI community building the future.* URL: https://huggingface.co/docs/transformers/model_doc/bert#overview.
37. Datasets. Hugging Face – The AI community building the future. URL: https://huggingface.co/docs/datasets/index.
38. DataSpeckle. What is an Attention Mask?. *LinkedIn:* Log In or Sign Up. URL: https://www.linkedin.com/pulse/what-attention-mask-dataspeckle/.
39. D-V-Murakhowsky - Overview. *GitHub*. URL: https://github.com/D-V-Murakhowsky.
40. Explore GitHub. *GitHub*. URL: https://github.com/explore.
41. Forbes Ukraine. *Telegram*. URL: https://t.me/Forbes_Ukraine_official.
42. FREEhost.UA. *Хостинг в Украине – купить украинский хостинг сайтов от провайдера FreeHost.* URL: https://freehost.com.ua/ukr/faq/wiki/chto-takoe-bazi-dannih-nosql/.
43. Gemini. *Google DeepMind.* URL: https://deepmind.google/technologies/gemini/.
44. genai. PyPI. URL: https://pypi.org/project/genai/.
45. Gensim: topic modelling for humans. *Radim Řehůřek: Machine learning consulting*. URL: https://radimrehurek.com/gensim/models/word2vec.html.
46. GitHub - lang-uk/tone-dict-uk: *Ukrainian tone dictionary.* GitHub. URL: https://github.com/lang-uk/tone-dict-uk.
47. GitHub - Oksana504/sentimentdictionary-uk. *GitHub*. URL: https://github.com/Oksana504/sentimentdictionary-uk.
48. GloVe: Global Vectors for Word Representation. *The Stanford Natural Language Processing Group*. URL: https://nlp.stanford.edu/projects/glove/.
49. Google AI for Developers | Build with the Google Gemini API and Gemma open models | *Google for Developers. Google for Developers*. URL: https://ai.google.dev/.
50. KoichiYasuoka/roberta-base-ukrainian-upos · Hugging Face. Hugging Face – The AI community building the future. URL: https://huggingface.co/KoichiYasuoka/roberta-base-ukrainian-upos.

51. Overview of ROBERTa model - GeeksforGeeks. GeeksforGeeks. URL: https://www.geeksforgeeks.org/overview-of-roberta-model/.
52. Overview. Stanza. URL: https://stanfordnlp.github.io/stanza/.
53. PyTorch. PyTorch. URL: https://pytorch.org/.
54. Reddit, Dive Into Anything. *Homepage - Reddit*. URL: https://www.redditinc.com/.
55. RoBERTa. Hugging Face – The AI community building the future. URL: https://huggingface.co/docs/transformers/model_doc/roberta.
56. Scikit-learn. *Scikit-learn: machine learning in Python – scikit-learn 0.16.1 documentation*. URL: https://scikit-learn.org/stable/.
57. Sentiment Analysis | Comprehensive Beginners Guide | *Thematic* | *Thematic*. *Thematic*. URL: https://getthematic.com/sentiment-analysis/#why-is-sentiment-analysis-important.
58. Shaikh R. Mastering BERT: A Comprehensive Guide from Beginner to Advanced in Natural Language Processing… Medium. URL: https://medium.com/@shaikhrayyan123/a-comprehensive-guide-to-understanding-bert-from-beginners-to-advanced-2379699e2b51.
59. Stack Overflow - Where Developers Learn, Share, & Build Careers. *Stack Overflow*. URL: https://stackoverflow.com/.
60. Streamlit. *A faster way to build and share data apps*. URL: https://streamlit.io/.
61. Telegram APIs. *Telegram APIs*. URL: https://core.telegram.org/.
62. Telethon's Documentation – Telethon 1.35.1 documentation. *Telethon's Documentation*. URL: https://docs.telethon.dev/en/stable/.
63. XGBoost Documentation – xgboost 2.0.3 documentation. *XGBoost Documentation*. URL: https://xgboost.readthedocs.io/en/stable/.

Додатки

Додаток 1

Тексти новин у форматі у форматі json та програмний код, який здійснює обробку та підготовку датасету. https://drive.google.com/drive/u/0/folders/1ja2KXivbiWHJeXJoF4jWJuHIZMZulnNc

Додаток 2

Програмний код для визначення тональності за допомогою тонального словника української мови та сам словник. https://drive.google.com/drive/u/0/folders/1XRr8-gEgm9fpZe4fp7XWkXoL3ImzauCb

Додаток 3

Програмний код для визначення тональності за ембедингами слів, що згенеровані локально розгорнутою мовною моделлю BERT, а також файл у форматі pkl, де зберегались отримані результати. https://drive.google.com/drive/u/0/folders/1AJO79WG10lNo03hbn4kTR1VOByPD3CIM

Додаток 4

Програмний код для визначення тональності за допомогою попередньо натренованої (fine-tuning) моделі Gemini-1.0-pro. https://drive.google.com/drive/u/0/folders/1nqBqza1mEl2aMHN9SgPpTMgVjwGIkwzh

Додаток 5

Програмний код для визначення тональності за допомогою попередньо натренованої моделі (fine-tuning) RoBERTa, а також код для застосунку. https://drive.google.com/drive/u/0/folders/1Spqze9o-WjLduhbavMqUXPC86vZVxm0V