НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

«На правах рукопису»

УДК _____

До захисту допущено:

в.о. завідувача кафедри
__Михайло НОВАТАРСЬКИЙ

«___»_____2025 р.

Магістерська дисертація

на здобуття ступеня магістра
за освітньо-науковою програмою «Комп'ютерні системи та мережі»
зі спеціальності 123 «Комп'ютерна інженерія»
на тему: «Метод визначення тональності текстових повідомлень за технологіями NLP»

Виконав:
студент ІІ курсу, групи ІО-31мн
Желепа Валентин Валерійович _____

Керівник:
проф. каф. ОТ, д.т.н.,
Писарчук Олексій Олександрович _____

Консультант з нормоконтролю:
проф. каф. ОТ, д.т.н.,
Кулаков Юрій Олексійович _____

Рецензент:
доцент каф. ОТ, д.н.т.,
Шимкович Володимир Миколайович _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ - 2025 року

Національний технічний університет України

«Київський політехнічний інститут імені ІГОРЯ СІКОРСЬКОГО»

Факультет (інститут) *Інформатики та обчислювальної техніки*
(повна назва)

Кафедра *Обчислювальної техніки*
Освітньо-кваліфікаційний ступінь *магістр*
(повна назва)
Спеціальність *123. Комп'ютерна інженерія*
(код і назва)
Освітньо-наукова програма *123. Комп'ютерні системи та мережі*
(код і назва)

ЗАТВЕРДЖУЮ
в.о. завідувача кафедри
Михайло Новотарський
(підпис) (ініціали, прізвище)
« » 2025 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Желепі Валентину Валерійовичу
(прізвище, ім'я, по батькові)

1. Тема дисертації *Метод визначення тональності текстових повідомлень за технологіями NLP*

Науковий керівник дисертації *проф. каф. ОТ, д.т.н. Писарчук О.О.*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від « 13 » березня 2025 р. № 1128-с

2. Строк подання студентом дисертації

3. Об'єкт дослідження *процес визначення тональності текстових повідомлень*

4. Предмет дослідження *алгоритми та методи класифікації тональності тексту технологіями NLP*

5. Перелік завдань, які потрібно розробити:

1. Аналіз особливостей інформаційного контенту інтернет-медіа.
2. Аналіз відомих методологічних підходів до визначення тональності текстових повідомлень за технологіями NLP.
3. Аналіз відомих технологічних рішень з визначення тональності текстових повідомлень за технологіями NLP.
4. Розробка методу визначення тональності текстових повідомлень за технологіями NLP.
5. Розробка програмної компоненти визначення тональності текстових повідомлень за технологіями NLP.

6. Консультанти розділів дисертації:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	проф. каф. ОТ д.т.н., Кулаков Ю.О.		
2	проф. каф. ОТ д.т.н., Кулаков Ю.О.		
3	проф. каф. ОТ д.т.н., Кулаков Ю.О.		

1. Дата видачі завдання _____01 вересня 2024_____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів дисертації	Примітка
1	*Затвердження теми*	*01.02.2025*	
2	*Вивчення літератури*	*06.02.2025*	
3	*Складання і узгодження технічного завдання*	*13.02.2025*	
3	*Написання вступної частини та огляд рішень*	*20.02.2025*	
4	*Моделювання розробленого способу*	*27.03.2025*	
5	*Оформлення документації ДП*	*06.04.2025*	
6	*Попередній захист та проходження нормативного контролю*	*05.05.2025*	
7	*Захист*	*19.05.2025*	

Студент _____ Валентин ЖЕЛЕПА
(підпис) (ініціали, прізвище)

Науковий керівник дисертації _____ Олексій ПИСАРЧУК
(підпис) (ініціали, прізвище)

РЕФЕРАТ
на магістерську дисертацію

виконану на тему: Метод визначення тональності текстових повідомлень за технологіями NLP

студентом: Желепою Валентином Валерійовичем

Робота складається із вступу та трьох розділів. Загальний обсяг роботи: 81 аркуша основного тексту, 8 ілюстрацій, 6 формул, 1 таблиця. При підготовці використовувалася література з 15 різних джерел.

Актуальність. У сучасному світі обсяг текстової інформації, який генерується щодня через Інтернет, соціальні мережі, блоги та інші цифрові платформи, зростає експоненціальними темпами. Це породжує як можливості, так і виклики для суспільства та наукової спільноти. Однією з ключових задач є розробка методів ефективного аналізу цієї інформації, що дозволяє не лише класифікувати дані, а й виділяти їх емоційну складову. Визначення тональності текстових повідомлень стає надзвичайно актуальним завданням, оскільки дозволяє оперативно оцінити ставлення споживачів, громадську думку, а також виявити потенційні ризики чи можливості для бізнесу та державного управління. Сучасні технології NLP, зокрема методи машинного і глибинного навчання, відкривають нові перспективи у точному аналізі тексту, проте їх практична адаптація часто потребує комплексного підходу та врахування специфічних особливостей оброблюваної інформації. Це дослідження є своєрідною відповіддю на потреби сучасного інформаційного суспільства, де швидкість прийняття рішень та точність аналізу даних стають вирішальними чинниками конкурентоспроможності підприємств. Використання інноваційних підходів у визначенні тональності тексту сприятиме розвитку як теоретичних основ NLP, так і їх практичному застосуванню у різних галузях – від маркетингу та фінансів до державного управління та безпеки. Актуальність теми підкріплюється зростаючим попитом на інтелектуальні системи аналізу даних, що забезпечують швидке реагування на зміни у настроях суспільства та ринкових умовах, а також сприяє розвитку інтегрованих систем підтримки прийняття рішень.

Зв'язок роботи з науковими програмами, планами, темами. Робота

інтегрується у загальнодержавні та кафедральні науково-дослідні програми, спрямовані на розвиток інтелектуальних технологій аналізу даних. Дисертаційне дослідження виконується у рамках проєкту, що підтверджують актуальність та практичну значущість запропонованих методів. Автор дисертації бере участь у виконанні зазначених наукових робіт шляхом написання магістерської дисертації.

Мета досліджень: підвищення оперативності та достовірності процесу визначення тональності текстових повідомлень за технологіями NLP.

Об'єкт дослідження – процеси обробки природньої мови з технологіями Natural Language Processing – NLP.

Предмет дослідження – методи машинного навчання для обробки природньої мови з технологіями Natural Language Processing – NLP.

Методи досліджень. У роботі використано метод машинного навчання, алгоритми обробки природної мови, а також методи валідації експериментальних даних. Кожен із зазначених методів застосовано для вирішення конкретних підзадач дослідження.

Наукова новизна. Новизна дослідження полягає у розробці інтегрованої методики, що поєднує традиційні підходи до аналізу тексту з сучасними моделями глибинного навчання. Отримані результати демонструють підвищення точності класифікації тональності порівняно з існуючими аналогами, що має значний потенціал для подальшого розвитку теорії та практики NLP.

Практичне значення. Запропонований метод успішно апробований в декількох інформаційних системах аналітики даних. Результати дослідження знаходять застосування в автоматизованих системах аналізу споживчих відгуків, моніторингу соціальних мереж та системах прийняття рішень в організаціях. Апробація здійснювалася у співпраці з провідними науковими установами та комерційними структурами.

Особистий внесок здобувача. Це магістерське дослідження являє особистий науковий внесок автора, включаючи унікальний підхід до роботи та самостійно отримані теоретичні та прикладні результати, що відповідають задачі оптимізації методу навчання з підкріпленням в контексті нейронних

мереж. Визначення цілі та завдань дослідження було виконано в співпраці з науковим керівником.

Ключові слова: Обробка природньої мови (NLP), тональний аналіз, емоційна оцінка тексту, машинне навчання, глибинне навчання, аналіз текстових даних, інтелектуальні системи, обробка даних, штучний інтелект, автоматизація аналізу тексту.

ABSTRACT

for a master's thesis

"Method for sentiment analysis of text messages based on NLP technologies"

author: Zhelepa Valentyn Valeriyovych

Scope and structure of the work. The dissertation is written on 81 pages. It contains a list of references to the sources used from 15 titles. The work contains 8 figures, 6 formulas and 1 table.

Relevance of the Topic. In the modern world, the volume of text information generated daily via the Internet, social networks, blogs and other digital platforms is growing exponentially. This creates both opportunities and challenges for society and the scientific community. One of the key tasks is to develop methods for effective analysis of this information, which allows not only to classify data, but also to highlight its emotional component. Determining the tone of text messages is becoming an extremely relevant task, as it allows you to quickly assess consumer attitudes, public opinion, as well as identify potential risks or opportunities for business and government. Modern NLP technologies, in particular machine and deep learning methods, open up new prospects in accurate text analysis, but their practical adaptation often requires a comprehensive approach and taking into account the specific features of the information being processed. This study is a kind of response to the needs of the modern information society, where the speed of decision-making and the accuracy of data analysis are becoming decisive factors of competitiveness. The use of innovative approaches in determining the tone of the text will contribute to the development of both the theoretical foundations of NLP and their practical application in various industries - from marketing and finance to public administration and security. The relevance of the topic is supported by the growing demand for intelligent data analysis systems that provide rapid response to changes in public sentiment and market conditions, and also contribute to the development of integrated decision support systems.

Connection with Scientific Programs. This study is integrated into national and departmental research programs aimed at advancing intelligent data analysis technologies.

Research Goal and Objectives. Improving the efficiency and reliability of the sentiment determination process for text messages using NLP technologies.

Object of Research. Processes of natural language processing using Natural Language Processing (NLP) technologies.

Subject of Research. Machine learning methods for natural language processing using Natural Language Processing (NLP) technologies.

Research Methods. The study employs machine learning methods, natural language processing algorithms, and experimental data validation techniques. Each of the mentioned methods is applied to solve specific sub-tasks of the research.

Scientific Novelty of the Results. The novelty of this research lies in the development of an integrated methodology that combines traditional text analysis approaches with state-of-the-art deep learning models. The results demonstrate a significant improvement in sentiment classification accuracy compared to existing methods, highlighting a substantial potential for further advancement in both the theory and practice of NLP.

Practical Significance of the Results. The proposed methodology has been successfully tested in various data analytics systems. The findings are applicable in automated systems for consumer review analysis, social media monitoring, and decision-support systems across organizations. The research results were validated in collaboration with leading scientific institutions and commercial entities.

Keywords: Natural language processing (NLP), tone analysis, emotional text evaluation, machine learning, deep learning, text data analysis, intelligent systems, data processing, artificial intelligence, automation of text analysis.

Пояснювальна записка

до магістерської дисертації

на тему:

«Метод визначення тональності текстових повідомлень за технологіями NLP»

Зміст

ЗАВДАННЯ ... 2

на магістерську дисертацію студенту ... 2

РЕФЕРАТ ... 4

на магістерську дисертацію ... 4

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ ... 12

ВСТУП ... 13

РОЗДІЛ 1 АНАЛІЗ ВІДОМИХ МЕТОДОЛОГІЧНИХ ТА ТЕХНОЛОГІЧНИХ ПІДХОДІВ ДО ВИЗНАЧЕННЯ ТОНАЛЬНОСТІ ТЕКСТОВИХ ПОВІДОМЛЕНЬ ЗА NLP-ТЕХНОЛОГІЯМИ ... 15

1.1. Особливості інформаційного контенту інтернет-медіа ... 15

1.2. Основні методологічні підходи та технологічні рішення з визначення тональності ... 16

1.3. Вітчизняні (українські) дослідження та їхній рівень ... 18

1.4. Відомі технологічні рішення для визначення тональності текстових повідомлень ... 22

1.5. Відомі методологічні підходи до визначення тональності тексту ... 23

1.5.1 Лексиконні (правила та словники) методи ... 23

1.5.2 Методи машинного навчання (класичні алгоритми) ... 26

1.5.3 Методи глибокого навчання (нейронні мережі) ... 29

1.5.4 Методи глибинного навчання (нейронні мережі) ... 32

1.5.5 Гібридні та ансамблеві підходи ... 36

1.6 Формалізація задачі дослідження та постановка часткових задач ... 37

ВИСНОВОК ДО РОЗДІЛУ 1 ... 40

РОЗДІЛ 2 РОЗРОБКА ПРОГРАМНОЇ КОМПОНЕНТИ ВИЗНАЧЕННЯ ТОНАЛЬНОСТІ ТЕКСТОВИХ ПОВІДОМЛЕНЬ ЗА ТЕХНОЛОГІЯМИ NLP 41

2.1. Розробка методу визначення тональності текстових повідомлень 41

2.2. Розробка програмної компоненти визначення тональності (архітектура) 46

2.2.1 Інженерія вимог 46

2.2.2 Архітектурне проєктування програмної компоненти 48

2.3. Програмна реалізація 53

ВИСНОВОК ДО РОЗДІЛУ 2 55

РОЗДІЛ 3 ПРИКЛАДНІ АСПЕКТИ ЗАСТОСУВАННЯ ПРОГРАМНОЇ КОМПОНЕНТИ ВИЗНАЧЕННЯ ТОНАЛЬНОСТІ ТЕКСТОВИХ ПОВІДОМЛЕНЬ ЗА ТЕХНОЛОГІЯМИ NLP 57

3.1. Демонстрація практичних можливостей розробки 57

3.2. Програмна документація на програмне забезпечення 63

ВИСНОВОК ДО РОЗДІЛУ 3 70

РОЗДІЛ 4 СТАРТАП: «ПРОГРАМНА КОМПОНЕНТА ВИЗНАЧЕННЯ ТОНАЛЬНОСТІ ТЕКСТОВИХ ПОВІДОМЛЕНЬ ЗА ТЕХНОЛОГІЯМИ NLP» 71

4.1. Опис стартап-проекту 71

4.2. Напрямки впровадження стартап-проекту 73

ВИСНОВОК ДО РОЗДІЛУ 4 77

ВИСНОВКИ 78

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 80

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ

NLP – Natural Language Processing (обробка природної мови)

TF-IDF – Term Frequency-Inverse Document Frequency (частота термів – обернена частота документа)

BoW – Bag-of-Words (мішок слів)

ШІ – штучний інтелект

МН – машинне навчання

ЛР – логістична регресія (Logistic Regression)

SGD – Stochastic Gradient Descent (стохастичний градієнтний спуск)

BERT – Bidirectional Encoder Representations from Transformers

RoBERTa – Robustly optimized BERT pretraining approach (модифікація моделі BERT)

SVM – Support Vector Machine (метод опорних векторів)

RNN – Recurrent Neural Network (рекурентна нейронна мережа)

LSTM – Long Short-Term Memory (рекурентна нейромережа з довгою короткочасною пам’яттю)

CNN – Convolutional Neural Network (згорткова нейронна мережа)

DNN – Deep Neural Network (глибока нейронна мережа)

SGD – Stochastic Gradient Descent (стохастичний градієнтний спуск)

API – Application Programming Interface (інтерфейс прикладного програмування)

HTML – HyperText Markup Language (мова розмітки веб-сторінок)

PEP – Python Enhancement Proposal (стандарт написання Python-коду)

VADER – Valence Aware Dictionary and sEntiment Reasoner (лексиконна модель аналізу тональності)

SGDClassifier – Stochastic Gradient Descent Classifier (класифікатор на основі стохастичного градієнтного спуску)

GPU – Graphics Processing Unit (графічний процесор)

MVP – Minimum Viable Product (мінімально життєздатний продукт)

SaaS – Software as a Service (програмне забезпечення як послуга)

ВСТУП

На сучасному етапі розвитку інформаційних технологій, зі зростанням щоденного обсягу текстової інформації в Інтернеті, соціальних мережах, електронних медіа досить чутливо обумовлюється нагальне практичне зацікавлення створенням автоматизованих методів аналізу цієї інформації. Важливим у цьому аспекті є напрям, що стосується визначення тональності текстових повідомлень. Цей процес базується на можливості класифікації текстів за характером емоційного прийому як позитивного, негативного або ж навіть нейтрального. Результати такого виду аналізу є великою цінністю для бізнесу, для маркетингу, для політичного аналізу і для систем підтримки рішень у державному управлінні.

Аналіз сучасної літератури дозволяє відзначити, що наукова спільнота приділяє значну увагу дослідженням у галузі обробки природної мови (NLP). Роботи багатьох дослідників зосереджені на розробці алгоритмів, які здатні ефективно інтерпретувати тональність текстів. Серед класичних підходів виділяються методи, що базуються на аналізі словникових ресурсів та статистичних показниках, проте їх точність часто обмежується у контексті обробки сучасних великих даних. Водночас, останні досягнення в області машинного та глибинного навчання, зокрема використання нейронних мереж, відкривають нові можливості для підвищення якості аналізу тональності. Ряд досліджень демонструє позитивну динаміку у впровадженні цих технологій, що дозволяє не лише адаптувати існуючі алгоритми до нових форматів даних, але й створювати інноваційні підходи для вирішення складних завдань.

За даними сучасних досліджень, серед яких варто виділити роботи провідних наукових груп у сфері NLP, стає очевидною необхідність інтеграції традиційних методів аналізу з новітніми технологіями штучного інтелекту. Такий інтегрований підхід сприятиме підвищенню точності класифікації текстових повідомлень, зокрема за рахунок більш глибокого розуміння контексту та семантичних особливостей тексту. Проте, незважаючи на успіхи, існує ряд відкритих питань, пов’язаних з обробкою неоднозначностей мови, адаптацією моделей до різних доменів та інтеграцією мультимодальних даних, що визначає подальшу перспективу досліджень у цій сфері.

Обґрунтування актуальності дослідження полягає в наступному. По-перше, зростання обсягів текстових даних і потреба в оперативній інтерпретації настроїв користувачів стимулюють розвиток нових методик аналізу, які здатні забезпечити високу точність та швидкодію. По-друге, успішна реалізація інтегрованих підходів до аналізу тональності сприятиме розвитку інтелектуальних систем підтримки прийняття рішень, що матиме практичне значення для широкого кола застосувань – від аналізу споживчих відгуків до моніторингу політичної ситуації. Нарешті, вирішення поставленої задачі дозволить зробити внесок у розвиток теоретичних основ NLP, розширюючи можливості сучасних моделей машинного навчання шляхом врахування специфічних особливостей емоційної оцінки тексту.

Таким чином, проведення даного дослідження є своєрідною відповіддю на сучасні виклики інформаційного суспільства. Воно спрямоване на розробку інтегрованої методики, що поєднує переваги традиційних підходів та сучасних технологій штучного інтелекту, з метою підвищення ефективності аналізу тональності текстових повідомлень. Отримані результати можуть стати важливою основою для подальших досліджень та практичного впровадження у сфері аналізу даних.

РОЗДІЛ 1

АНАЛІЗ ВІДОМИХ МЕТОДОЛОГІЧНИХ ТА ТЕХНОЛОГІЧНИХ ПІДХОДІВ ДО ВИЗНАЧЕННЯ ТОНАЛЬНОСТІ ТЕКСТОВИХ ПОВІДОМЛЕНЬ ЗА NLP-ТЕХНОЛОГІЯМИ

Аналіз тональності тексту (сентимент-аналіз, opinion mining) – це сучасний напрям комп'ютерного «мовознавства», що вивчає процес визначення, виявлення та класифікації емоційного забарвлення текстів, або ж навіть відчуттів що намагвся донести автор при його написанні. Просто кажучи, дана область вивчення має на меті визначити в якому стані автор висловлює ставлення до об'єкта або події, зазначеного у повідомленні. Ця галузь набирає великої популярності з поширенням соціальних мереж типу «Х», месенджерів «Телеграм», відгуків покупців аналітичних додатків, новин тощо, де вимірювання характеру і відчуттів публіки мають прикладну користь у маркетингу, соціології, політології тощо. Підсумовуючи, за останні роки методи сентимент-аналізу пройшли шлях від звичайних простих лексичних підходів до дійсно складних моделей глибинного навчання, що призвело до значного підвищення точності і масштабованості аналізу. У розібраному далі розділі представлено існуючі у світі методи аналізу тональності текстів, як світових, так і українських напрацювань в цій галузі, та проведено критичний аналіз їхніх переваг, недоліків і перспектив розвитку.

1.1. Особливості інформаційного контенту інтернет-медіа

Інтернет-ресурси генерують насправді величезні обсяги текстового контенту, значна частина якого створюється безпосередньо користувачами на кшталт нас з вами (User-Generated Content, UGC) та безперервно публікується, постійно оновлюючись та додаючи щось нове [1]. Такий інформаційний потік включає як новинні статті, блоги, соціальні мережі, так і звичайні коментарі. Для привернення уваги аудиторії великі інтернет-видання часто використовують емоційно забарвлені заголовки і тексти, оскільки це здається більш привабливим та цікавим людині. Думаю всі помітили, що з часом заголовки новин стають більш поляризованими (в тому числі політично) та емоційно насиченими, оскільки редакції прагнуть максимізувати клікабельність (CTR) та залучення читачів, бо кількість читачів – це їх

безпосередній хліб. Зокрема, аналіз 23 млн новинних заголовків у США виявив поступове зростання негативності та частоти емоцій (злість, страх, відраза тощо) в заголовках протягом 2000–2019 років. Інформаційний контент інтернет-медіа має помітний вплив на суспільну думку, а дослідження доводять, що тональність новинного контенту може слугувати індикатором громадського настрою та думок [2]. Водночас тексти в інтернеті часто містять неформальну лексику, сленг, емодзі та інші особливості, особливо в соціальних мережах, де щодня у молоді створюються нові меми та сленгові вирази з поп-культури. Це створює виклики для автоматичного аналізу тональності, адже системі потрібно правильно інтерпретувати не тільки літературну мову, але й інтернет-жаргон, сарказм та контекстні натяки. Таким чином, особливостями інтернет-контенту є: величезний обсяг і швидкість появи нових даних; часто емоційно забарвлена подача інформації (іноді тенденційно негативна для привернення уваги); наявність розмовних та неформальних елементів мови; різноманітність жанрів (від новин до твітів) і, як наслідок, різна стильова та мовна специфіка.

1.2. Основні методологічні підходи та технологічні рішення з визначення тональності

Аналіз тональності (Sentiment Analysis) — одна з найбільш актуальних галузей у сфері обробки природної мови (Natural Language Processing, або скорочено NLP). Вона набула значного розвитку завдяки стрімкому розвитку технологій штучного інтелекту та глибокого навчання, бо саме ці технології значно спростили дослідження, які до цього проводились вручну. Основними методологічними підходами визначення тональності можна виділити словникові (лексиконні), методи машинного навчання та гібридні підходи.

Словникові підходи базуються на спеціалізованих лексиконах, які вже містять слова з визначеним емоційним навантаженням, позначені певним числовим значенням, що у свою чергу характеризує тональність. Числові значення для програм є куди зрозумілішими ніж слова, тому саме від числового значення в подальшому класифікують тональність. Типовим прикладом є використання лексиконів SentiWordNet та VADER, які дозволяють ефективно здійснювати простий аналіз полярності на рівні слів і фраз. Перевагою цих

методів є доволі гарна швидкість роботи та відсутність потреби у навчанні на якихось великих датасетах, але варто зазначити, що вони не завжди здатні точно врахувати контекст речень та складні мовні конструкції, наприклад сарказм або іронію.

У свою чергу, методи машинного навчання передбачають створення моделей, навчених на попередньо розмічених датасетах. Починаючи з класичних методів (звичайна логістична регресія, наївний байєсівський класифікатор, або ж метод опорних векторів (SVM)), у сучасних дослідженнях перевага віддається нейронним мережам, зокрема згортковим нейронним мережам (CNN), рекурентним нейронним мережам (RNN, LSTM) та моделям на основі трансформерів (наприклад, BERT, RoBERTa) [1]. Ці моделі вже дозволяють враховувати глибокий контекст і показують значну точність на великих наборах даних. Наприклад, вищезазначена модель BERT значно підвищила ефективність аналізу тональності текстових повідомлень, оскільки здатна до глибокого розуміння семантики тексту, забезпечуючи точність понад 90% на багатьох еталонних наборах даних, таких як IMDb (відгуки на фільми), Yelp (відгуки на ресторани) та Amazon Reviews (відгуки товарів з онлайн-магазину).

Гібридні ж методи поєднують як лексиконні підходи так і методи машинного навчання для спроби підвищення точності аналізу тональності. Наприклад, на першому етапі може використовуватися словниковий метод для швидкого визначення базової тональності тексту, після чого на другому етапі моделі глибокого навчання уточнюють ці оцінки, враховуючи вже складніші контекстуальні залежності [1]. Такі гібридні рішення демонструють синергічний ефект та забезпечують високу точність аналізу, особливо у випадках текстів зі складною семантичною структурою або наявністю сарказму. Беручи найкраще з двох методів можна досягти ліпшого кінцевого результату та усунути недоліки кожного з методів окремо, але такий підхід є дорогим у ресурсах, та застосовується зазвичай лише компаніями у великих комерційних проектах.

Останні технологічні рішення у сфері визначення тональності характеризуються також розширенням самого спектру завдань аналізу

тональності. Окрім класичної для всіх бінарної класифікації («позитивне-негативне») активно розвиваються аспектно-орієнтований аналіз тональності (ABSA), що дозволяє аналізувати тональність щодо окремих аспектів продукту чи послуги, та аналіз емоцій (Emotion Analysis), який дозволяє визначати конкретні емоції (наприклад, радість, сум, гнів) у текстових повідомленнях. Схожу технологію застосовує популярний ресурс Grammarly паралельно з граматичною перевіркою англомовних текстів.

Також варто відзначити тенденцію мультимодального аналізу тональності, який враховує не лише текстову інформацію, а й візуальні та навіть аудіо-компоненти повідомлень. Це дозволяє отримувати точнішу оцінку емоційної забарвленості повідомлень у соціальних мережах, де часто текст супроводжується зображеннями або аудіозаписами.

Таким чином, аналіз тональності за технологіями NLP сьогодні є багатогранною задачею, яка інтегрує різні методи, підходи та технологічні рішення, забезпечуючи високий рівень точності та адаптивності до різноманітних практичних завдань, що постають перед розробниками.

1.3. Вітчизняні (українські) дослідження та їхній рівень

Розвиток аналізу тональності українською мовою тривалий час відставав від англомовного сегменту, що зумовлено як меншою кількістю доступних даних, так і меншою увагою дослідників до цієї прикладної задачі в минулому. Проте за останні ~5 років ситуація помітно покращилася: підвищився інтерес до автоматичної обробки українських текстів у цілому, з’явилися локальні набори даних і перші конкурентоспроможні моделі.

Одним з перших кроків стало створення тональних словників для української мови. Прості списки слів з позначками тональності існували ще з 2010-х років (переважно перекладені або адаптовані з англійських джерел чи російських словників тональності). У 2018 році було опубліковано Ukrainian Sentiment Lexicon на платформі Figshare – близько 2822 українських слова, оцінені за шкалою від -5 до +5. Цей словник (іноді його називають "Полярний Український словник 2014", оскільки збирався після подій 2014 року) надав дослідникам базовий інструмент для побудови словникових систем. Паралельно ентузіасти на GitHub і Kaggle публікували зібрані списки

позитивної/негативної лексики. Хоча ці ресурси не такі великі, як, скажімо, англомовний SentiWordNet, вони стали відправною точкою для українських лексичних методів.

Стосовно корпусів розмічених даних, довгий час їх майже не було. Лише поодинокі дослідники вручну формували невеликі набори відгуків чи коментарів. Ситуація почала змінюватися після 2019 року. Наприклад, на міжнародній платформі Kaggle стали з'являтися набори українських текстів з мітками настрою – зокрема, вибірки твітів з хештегами, пов'язаними з війною 2022 року, де твіти розмічені як проукраїнські чи проросійські за тональністю, а також дані з Facebook/Telegram коментарів, розмічені волонтерами. У 2020 році українські науковці вперше масово представили роботи на локальних конференціях з тематичними наборами: UA Review Corpus – наприклад, набір відгуків про послуги або товари українською з тональними оцінками. У 2021 році на конференції ICTERI було презентовано дослідження Панченко та ін. про оцінку різних моделей NLP для української класифікації текстів, де серед задач була і класифікація тональності. Вони зібрали так званий "Model Zoo" – набір різних моделей (від класичних до BERT) – і порівняли їх на спільному корпусі, показавши, які технології дають кращий результат для українських текстів.

Окремо слід відзначити проект Харківського національного університету радіоелектроніки (ХНУРЕ). В 2021 році дослідники ХНУРЕ (Рябишев О. та ін.) опублікували статтю «Аналіз тональності тексту українською мовою», де здійснили комплексне дослідження методів і розробили власний корпус. Вони автоматично зібрали відгуки користувачів про мобільні додатки з Google Play (близько 6000 відгуків) і розмітили їх на позитивні та негативні. Цей датасет став одним із найбільших на той час для української. Далі автори протестували кілька алгоритмів бінарної класифікації – зокрема, методи на основі ознак (ймовірно Naive Bayes, логістична регресія тощо) та нейронні мережі – щоб знайти оптимальний підхід. Результати показали, що попередньо навчена багатомовна модель BERT продемонструвала найвищу точність у визначенні тональності українських текстів. Власне, їм вдалося успішно розв'язати задачу

класифікації відгуків як позитивних/негативних, побудувавши модель на основі Multilingual BERT, яка перевершила інші алгоритми на їх вибірці. Ця робота важлива тим, що вперше явно показала ефективність трансформерів для української мови і підтвердила, що сучасні підходи можуть бути перенесені на українські дані.

У 2020–2022 рр. з'явилася низка праць українських науковців, присвячених саме тональності: Шаховська та ін. (Львівська політехніка) у 2020 р. опублікували модель аналізу тональності відгуків споживачів (наприклад, сервісів) з використанням нейронних мереж, результат якої ~88% точності. Згідно з їх роботою, було використано комбінацію методів – імовірно, поєднання класичного підходу (регресії, SVM) з правилом – для досягнення такого результату на власноруч зібраних даних з Google Play. Інша робота (Оленич та ін., 2021) представила систему автоматичного визначення тональності українського тексту, апробовану на матеріалах новин. У 2022–2023 рр. дослідники з Хмельницького нац. університету (Залуцька О. та ін.) зосередилися на аналізі тональності у сфері електронної комерції. В їх праці 2023 року (конференція COLINS) запропоновано метод аналізу тональності відгуків українською мовою з використанням модифікованої моделі RoBERTa. Автори сформували корпус коротких відгуків "суржиком" (тобто реальні відгуки, де українська перемішана зі сленгом і росіянізмами) і навчили модель, досягнувши 92% точності у бінарній класифікації. Важливо, що метод працював навіть на текстах, де не дотримано літературних норм (мішанина мов, помилки) – такого роду виклики притаманні соцмережам і відгукам, і модель на основі RoBERTa виявилася досить стійкою. Цей результат майже відповідає рівню найкращих англомовних моделей для аналогічних задач, що свідчить: при наявності ресурсів (гарний корпус і сучасна модель) український сентимент-аналіз може вийти на світовий рівень.

Попри такі успіхи, загалом вітчизняні дослідження ще знаходяться на етапі наздоганяння**.** Проблемним питанням залишається дефіцит великих та безпосередньо публічних датасетів**.** Більшість корпусів, згаданих вище, зібрані одноразово для конкретних проєктів і не завжди доступні відкрито. Для

порівняння, в англомовному середовищі існують десятки загальнодоступних корпусів (IMDB, Yelp, Amazon, Twitter тощо) та щорічні змагання (SemEval, Kaggle), що рухають прогрес. Українська ж мова є менш забезпеченою: великий набір TBCOV (Tweets by COVID-19) 2021 року, що містить 2 мільярди твітів з різними мовами, включає лише ~3400 твітів українською з позначками тональності, тобто частка українського матеріалу мізерна. Інший аспект – особливості мови: українська – синтетична, з багатою морфологією, і часто користувачі мереж пишуть змішаним стилем, використовуючи діалектизми, росіянізми, або чергуючи українську та інші мови в одному тексті. Навіть я сам так часто роблю, оскільки сучасна інтернет культура дуже глибоко пов'язана з іноземним елементом. Це ускладнює застосування моделей, які не пристосовані до code-switching. Деякі дослідження відзначають, що через недостатню формалізацію української в інтернеті моделі можуть плутатися у визначенні мови та залученні відповідних тональних ресурсів [7].

Відповідно рівень українських напрацювань трохи нижчий: якщо англомовні моделі давно інтегрують, скажімо, виявлення сарказму або аспектний аналіз у бізнес-застосунках, то українські поки що зосереджені переважно на базовій класифікації «позитив/негатив» і поступово розширюють охоплення (з’являються перші роботи про тональність в українських соцмережах, про аналіз емоційності коментарів тощо).

Порівнюючи рівень: точність сучасних українських моделей, як було згадано, може досягати ~90%+ на спеціалізованих завданнях, що співмірно з англомовними. Але різниця в доступності: англійською є готові переднавчені моделі (наприклад, ToneAnalyzer від IBM, або моделі в HuggingFace, навчені на мільйонах твітів), натомість для української якісні моделі (наприклад, український BERT) лише починають з’являтися і потребують доопрацювання. Останнім часом, втім, спостерігається позитивний рух: з огляду на зростання ролі української мови в інтернеті та підтримку з боку державних програм з розвитку українського мовного технологічного простору, можна очікувати, що кількість і якість ресурсів стрімко зростатиме.

1.4. Відомі технологічні рішення для визначення тональності текстових повідомлень

На сьогодні існує багато готових технологічних рішень, сервісів та інструментів, які дозволяють визначати тональність тексту з використанням методів NLP. Вони варіюються від хмарних API великих технологічних компаній до відкритих бібліотек та спеціалізованих програмних продуктів. Розглянемо кілька прикладів:

- ***Хмарні NLP-сервіси:*** Провідні IT-компанії пропонують API для аналізу тональності. *Google Cloud Natural Language API* визначає sentiment score (полярність від -1.0 до 1.0) та magnitude (інтенсивність емоцій) для тексту на рівні документа або окремого речення. *IBM Watson Natural Language Understanding (NLU)* використовує глибоке навчання для аналізу тексту і, окрім тональності (полярність), може виявляти емоційні тони – наприклад, смуток, радість, страх, відразу, гнів. *Amazon Comprehend* та *Azure Cognitive Service for Language* надають схожі можливості: так, Azure визначає тональність як позитивну, нейтральну, негативну чи змішану, з відповідними коефіцієнтами впевненості, і підтримує 94 мови, включно з опцією виділення думок про окремі аспекти (Opinion Mining) [3]. Використання цих хмарних рішень вимагає мінімум власних зусиль на навчання моделей – вони надають готові попередньо навчені моделі, доступні через API.

- ***Інструменти для соціальних медіа та маркетингу:*** Багато платформ моніторингу соцмедій мають вбудований аналіз тональності. Наприклад, *Sprout Social* – платформа керування соцмережами – автоматично класифікує повідомлення та коментарі як позитивні, негативні чи нейтральні, і дозволяє відстежувати середній рівень тональності аудиторії бренду з часом. Подібно, сервіси на кшталт *Hootsuite, Meltwater* тощо інтегрують аналіз тональності для брендів, включаючи візуалізацію трендів та метрик тональності в дашбордах.

- ***Відкриті бібліотеки та моделі:*** Для розробників доступні численні open-source бібліотеки Python для аналізу настроїв. *NLTK* (Natural Language Toolkit), який містить словники сентименту, такі як *VADER (скорочено від Valence Aware Dictionary and sEntiment Reasoner)* – лексикон та набір правил,

спеціально налаштований для мови соцмереж [4]. VADER визначає полярність і інтенсивність висловлювань (з урахуванням сленгу, емодзі, пунктуації) без потреби навчання на корпусі, тому є популярним для аналізу твітів та відгуків. *TextBlob* – ще одна Python-бібліотека, що пропонує простий інтерфейс для аналізу тональності (вона фактично використовує правило- та лексикон-орієнтований підхід, аналогічний до VADER). Більш сучасні варіанти – *бібліотеки глибокого навчання*: наприклад, HuggingFace Transformers дозволяють завантажити готові моделі на зразок BERT, спеціально донавчені для sentiment analysis. Існують багатомовні моделі (наприклад, mBERT, XLM-RoBERTa) та навіть моделі для української мови (наприклад, UkroBERT), які можна використати для класифікації тональності українських текстів. Однак моделі на основі Transformer великі за розміром і потребують більше ресурсів для інференсу, тому їх використання виправдане, коли потрібна максимальна точність.

Таким чином, на ринку доступний широкий спектр технологічних рішень: від простих у використанні хмарних сервісів (Google, IBM, Microsoft тощо) до кастомізованих інтегрованих платформ і відкритих бібліотек для самостійної розробки. Вибір рішення залежить від вимог до точності, мовної підтримки, обсягу даних та обчислювальних ресурсів.

1.5. Відомі методологічні підходи до визначення тональності тексту

Існуючі методології аналізу тональності (sentiment analysis) умовно поділяються на кілька груп: лексиконні (словникові) методи, методи машинного навчання (ML, зокрема глибокого навчання) та гібридні методи, що поєднують перші два [5]. Кожен підхід має свої переваги і недоліки, а вибір залежить від наявності навчальних даних, необхідної глибини аналізу та ресурсних обмежень.

1.5.1 Лексиконні (правила та словники) методи

Лексиконні методи здійснюють аналіз тональності, опираючись на заздалегідь підготовлені словники тональних лексем (слова/фрази зі вказаною полярністю) та набір правил. Кожному слову в такому словнику призначено певний тональний вага або оцінка (позитивна, негативна, нейтральна, іноді

числовий бал інтенсивності). При обробці тексту алгоритм знаходить слова із словника та обчислює загальну тональність повідомлення – наприклад, шляхом сумування полярностей або визначення переважного тону. Додатково враховуються прості лінгвістичні правила, такі як інверсія тональності при наявності заперечень (“не добрий” → негативний тон) чи посилення/послаблення тональності за допомогою прислівників-інтенсифікаторів (“дуже добрий” → підсилення позитивної оцінки).

У світовій практиці створено кілька відомих лексиконів тональності: SentiWordNet**,** AFINN**,** VADER**,** LIWC тощо. Наприклад, VADER (Valence Aware Dictionary and sEntiment Reasoner) – це популярний лексиконно-правиловий алгоритм, спеціально налаштований для аналізу тональності повідомлень у соціальних мережах (враховує розмовні скорочення, емотикони, інтернет-сленг). Дослідження показують, що VADER здатен досить якісно визначати тональність коротких текстів; зокрема, було відзначено його перевагу в точності над простішими методами типу TextBlob при аналізі твітів [6]. Інший приклад — алгоритм SO-CAL (Semantic Orientation Calculator), що використовує словники з оцінками полярності та враховує граматичні конструкції для підсилення/негування; було показано, що він забезпечує стабільні результати на різних доменах і навіть на текстах, не задіяних під час створення словника. Для української мови також створюються словники тональності (наприклад, український словник полярних прикметників та прислівників), хоча їх масштаб поки що менший, ніж в англійській.

Головна перевага – простота та незалежність від тренувальних даних. На відміну від машинного навчання, для використання словникового методу не потрібна попередня розмітка великого корпусу і навчання моделі. Достатньо мати готовий словник і набір правил, після чого можна одразу аналізувати довільні тексти. Це робить лексиконні методи надзвичайно швидкими та зручними у впровадженні [7]. Також вони легко інтерпретуються: можна зрозуміти, чому текст отримав певну оцінку (напр., через наявність конкретних “позитивних” чи “негативних” слів). Лексиконні методи зазвичай добре працюють між доменами, оскільки базові списки слів загальної мови можуть

застосовуватися до різних тематик без додаткового навчання [8]. Більше того, лексиконний підхід дозволяє визначати градації тональності – наприклад, обчислювати сумарний “бал сентименту” тексту на континуумі від дуже негативного до дуже позитивного. Така тонка оцінка може давати глибші інсайти та пояснення, ніж просто позитив/негатив, що цінно для соціальних наук та маркетингу, де важливо відстежувати зміни настроїв.

Основний недолік – відносно нижча точність на складних текстах у порівнянні з сучасними методами машинного навчання. Лексикон часто не враховує контекст слів. Наприклад, слово *“зарядка”* може мати позитивну конотацію (зарядка для тіла – корисно) або нейтральну (зарядка пристрою) залежно від контексту; словник же має фіксовану оцінку і не “розуміє” контекст. Також такі методи погано розпізнають сарказм, іронію, приховані смисли – фраза *“Ну просто чудово!”* може вживатися зі справді позитивним або саркастично негативним значенням, і без контексту словниковий метод це не розрізнить. Інша проблема – доменно-специфічна лексика: загальні словники можуть не містити спеціальних термінів або сленгу певної галузі. У текстах, наприклад, з IT-форумів чи медичних відгуків, можуть бути слова, відсутні у загальному словнику, або звичні слова, що мають специфічну тональність у цьому домені. Лексиконний метод потребує або розширення словника для нового домену, або стикається зі зниженням якості класифікації. Автоматична побудова чи адаптація словників – нетривіальне завдання; потрібно враховувати, що значення слів можуть змінюватися з часом (мовна еволюція, появу нового сленгу). У дослідженнях відзначається, що для забезпечення високої ефективності лексиконного аналізу часом необхідно автоматично доповнювати словники доменно-специфічною лексикою та вагами, що ускладнює процес [6].

Не зважаючи на ці обмеження, лексиконні методи продовжують широко застосовуватися, особливо коли необхідна швидка оцінка тональності великих масивів текстів або потрібна прозорість у прийнятті рішення алгоритмом. Ба більше, сучасні дослідження пропонують вдосконалення цього підходу. Наприклад, метод MultiLexScaled, що усереднює оцінки з кількох словників,

показав досить високу узгодженість із людською розміткою на різних наборах даних, подекуди наближаючись за точністю до моделей машинного навчання. Отже, лексиконний підхід забезпечує базову платформу для аналізу тональності, яка може слугувати або самостійним швидким рішенням, або компонентом складніших гібридних систем.

1.5.2 Методи машинного навчання (класичні алгоритми)

Методи цієї групи використовують алгоритми класичного (статистичного) машинного навчання для класифікації текстів за тональністю. На відміну від лексиконного підходу, тут модель навчається на розмічених даних – великій вибірці текстових документів, кожен з яких має відомий клас тональності (позитивний, негативний, нейтральний). В процесі навчання модель визначає патерни (співвідношення характеристик тексту і його тональності) і надалі застосовує їх для прогнозування тональності нових повідомлень. Ключовим етапом є представлення тексту у вигляді числових ознак (features), придатних для алгоритму навчання. Типово використовуються підходи bag-of-words (мішок слів) та похідні від нього: будується словникове представлення, де кожне слово (або n-грама слів) відповідає певному виміру простору ознак, і текст перетворюється на вектор частот або ваг слів. Популярною є схема зважування TF-IDF (term frequency–inverse document frequency), що підвищує значущість рідкісних слів і зменшує вплив надто частотних [9]. Також можуть додаватися спеціалізовані ознаки: кількість позитивних/негативних слів (за словником), наявність знаків оклику чи емотиконів, довжина тексту, тощо. На такому векторному поданні даних застосовуються стандартні алгоритми класифікації: наївний Байєс (NB), логістична регресія (LR), метод опорних векторів (SVM), дерева рішень (C4.5, CART), k-ближніх сусідів (k-NN), ансамблеві методи (Random Forest, AdaBoost тощо). Ці алгоритми протягом багатьох років були основою автоматичної класифікації текстів і зарекомендували себе як надійні інструменти [10].

Класичні моделі здобули популярність у перших дослідженнях sentiment analysis і досі широко використовуються як базові орієнтири. Зокрема, алгоритм наївного Байєса було застосовано ще в одній з перших робіт по

аналізу тональності кіно-рецензій; його точність виявилася цілком пристойною, що підтвердило життєздатність машинного підходу. Надалі SVM та логістична регресія часто демонстрували найкращу якість серед класичних методів на різних корпусах. Наприклад, у дослідженні Madhuri (2019) на задачі класифікації тональності твітів про залізницю Індії, SVM досяг точності ~91,5%, випередивши дерева рішень, Random Forest та Bayes-класифікатор (у яких точність була близько 89%). Інше дослідження, присвячене тональності твітів про вибори в Індонезії, навпаки показало кращий результат наївного Байєса (75,6%), порівняно з SVM (64%) та k-NN (73%) – це свідчить, що оптимальний вибір алгоритму може залежати від специфіки даних і розподілу ознак. В цілому, на великих і збалансованих корпусах класичні методи здатні давати 70–85% точності. Наприклад, байєсівський класифікатор, навчений на 1,6 мільйонах твітів (Sentiment140), показував ~85% точності на тестових даних. Лінійні моделі (логістична регресія, Ridge regression) на відгуках IMDb досягають ~90% точності, що наближається до результатів більш складних моделей. Таким чином, навіть без глибоких нейронних мереж класичне ML може забезпечити досить високий рівень правильного розпізнавання тональності [9].

У класичному методі машинного навчання є очевидні деякі переваги. По-перше, він гнучкий і адаптивний, бо модель можна навчити під конкретний домен або навіть конкретну мову, маючи відповідні розмічені дані. Це дозволяє врахувати специфічну лексику, жаргон, стиль – усе, що може не бути в загальних словниках. По-друге, класичні алгоритми відносно ефективні обчислювально – навчання і виконання (класифікація нових зразків) зазвичай відбувається швидше і потребує менше ресурсів, ніж у глибоких нейронних мереж. Їх можна запускати на звичайному CPU без спеціалізованого обладнання. По-третє, вони часто забезпечують прийнятну інтерпретованість: наприклад, ваги ознак у логістичній регресії чи важливості ознак у дереві рішень можуть дати уявлення, які слова найбільше впливають на результат класифікації. Це корисно при аналізі моделі та довірі до її рішень (хоча деталізація не така проста, як у лексиконних правил). Нарешті, для реалізації

класичних методів існує багато готових бібліотек та інструментів (scikit-learn, NLTK тощо), що спрощує впровадження.

Ключовий недолік – потреба у достатньо великому та попередньо розміченому корпусі даних для навчання. Збирання і розмітка (анотування тональності) текстів – трудомісткий процес, особливо для специфічних доменів чи мов, де немає готових датасетів. Якщо даних мало або вони зміщені (unbalanced), якість навченої моделі може бути низькою. Крім того, класичні алгоритми зазвичай оперують поверхневими ознаками (окремими словами чи n-грамами) і не враховують глибинний контекст словосполучень. В результаті модель може помилятися на складних випадках, де значення фрази залежить від далекого контексту або синтаксичної структури. Наприклад, модель може неправильно класифікувати речення з запереченням, якщо комбінація слів не була достатньо представлена у навчальних даних. Щоб частково це врахувати, застосовують n-грами, але це обмежено короткими дистанціями. Також класичні моделі схильні бути доменно-залежними: модель, навчена на відгуках про фільми, може не дати такого ж результату на, скажімо, відгуках про ресторани, якщо стилістика і словник відрізняються. Потрібне перенавчання або адаптація до нового домену. Ще одна проблема – обмежена здатність моделювати складні мовні феномени. Сарказм, переносні значення, контрастивні конструкції (напр. “Хоч продукт і дешевий, але якість чудова”) можуть заплутати алгоритм, якщо явно не додати спеціальних правил або ознак.

Попри це, класичні ML-методи залишаються популярними, та є важливою частиною сучасних систем аналізу тональності. Вони часто використовуються як базова лінія для порівняння з новими підходами. Крім того, їх включають до складу ансамблевих моделей: наприклад, поєднання кількох різних класифікаторів (NB, SVM, LR) голосуванням може дати кращий результат, ніж кожен з них окремо. Такі ансамблі підвищують надійність результату і знижують вплив вдалих/невдалих випадків окремого алгоритму.

1.5.3 Методи глибокого навчання (нейронні мережі)

Наступним етапом у розвитку аналізу тональності стало впровадження глибоких нейронних мереж (Deep Learning). Близько 2013–2015 років, з піднесенням глибокого навчання в комп'ютерному зорі і розпізнаванні мови, дослідники NLP почали експериментувати з нейромережевими моделями для задач текстової класифікації, включно з sentiment analysis. Основна ідея – використовувати нейронні мережі для автоматичного виокремлення ознак з тексту, що дозволяє відійти від вручну сконструйованих ознак (як у класичних методах). Нейронні мережі здатні навчитися складнішим шаблонам і враховувати ширший контекст слів у реченні. Застосовуються різні архітектури: згорткові нейронні мережі (CNN) – добре виділяють локальні патерни (наприклад, фрази характерного забарвлення), рекурентні нейронні мережі (LSTM, RNN та GRU) – моделюють послідовні залежності та довготривалі зв'язки у тексті. Входом до нейронної мережі зазвичай служать не просто частоти слів, а їх векторні представлення (embedding) – щільні числові вектори, що відображають семантичне значення слів. Такі вектори можуть бути попередньо навчені на великих корпусах (наприклад, популярні моделі Word2Vec, GloVe, fastText дають сталий вектор для кожного слова). Використання ембедінгів дозволяє мережі простіше узагальнювати синонімію та схожість слів. Після перетворення слів у ембедінги, нейронна мережа (CNN або LSTM тощо) генерує з усього тексту певне приховане представлення, яке потім через вихідний шар класифікується в класи тональності. Таким чином, глибока модель сама вчиться оптимальних ознак і правил класифікації, мінімізуючи помилки на навчальній вибірці.

Глибокі моделі швидко продемонстрували суттєве підвищення точності на багатьох задачах аналізу тональності. Вже перші результати показали, що нейронні мережі можуть перевершувати класичні алгоритми, особливо на великих обсягах даних. Для ілюстрації, CNN-модель, навчена на відгуках IMDb (кілька тисяч прикладів), досягла точності ~99,3%, перевершивши всі . Рекурентні мережі (LSTM) особливо ефективними виявилися на даних з довгими залежностями – наприклад, на великому корпусі твітів LSTM змогла

підняти точність до ~82%, тоді як найкращий класичний алгоритм ледве досягав ~75%. Багатошарові перцептрони (MLP) також застосовувалися: на турецьких твіттах MLP дав ~81,9% точності, на англомовних COVID-твітах глибока мережа з 5 шарами досягла 93,7%. Таким чином, вихід на рівень 80–90%+ точності для задач тональності значною мірою став можливим завдяки глибоким нейромережам. Вони краще засвоюють контекст: якщо класичному алгоритму важко врахувати далекі заперечення чи зміну тону в кінці речення, то LSTM, пропускаючи послідовність через свою пам’ять, краще справляється з такими випадками.

Серед практично успішних підходів можна назвати модель CNN, запропоновану Кімом (Y. Kim, 2014) для класифікації речень – проста одношарова згортка над ембедінгами слів дала сильний базовий результат на кількох датасетах тональності, ставши відправною точкою для наступних робіт. RNN/LSTM моделі були використані для аналізу тональності оглядів товарів, твітів та навіть для аспект-орієнтованого аналізу тональності (визначення тональності щодо конкретних аспектів продукту). Комбінації моделей теж дають ефект: наприклад, гібрид CNN-LSTM може спочатку виокремити локальні особливості фраз, а потім узагальнити їх послідовність – такий підхід показав високі результати на змішаних задачах [9].

Нерідко застосовують і ансамблі нейронних мереж або їх поєднання з класичними методами (наприклад, середнє або голосування прогнозів декількох різних нейронних архітектур). У 2010-х роках глибокі мережі встановили нові рекорди точності на стандартних наборах: для кіно-рецензій IMDb перевищено 95%, для твіттер-корпусу Sentiment140 досягнуто ~85%+, для Yelp-відгуків – понад 90% тощо. Ці успіхи закріпили положення глибокого навчання як провідного інструмента в аналізі тональності.

Головна перевага такого методу – вища якість класифікації завдяки здатності моделі навчитися складних нелінійних залежностей. Нейронна мережа може врахувати контекст слова, позицію в реченні, навіть певні синтаксичні структури (напряму чи опосередковано). Вона автоматично генерує ознаки на основі даних, що знімає тягар фічерінженірингу з

розробника: не потрібно вручну придумувати тисячі правил чи ознак, достатньо забезпечити модель даними і обчислювальні ресурси для навчання. До того ж, векторні представлення слів дозволяють моделі узагальнювати знання: навіть якщо конкретне слово нечасто траплялося у тренуванні, але має схожий embedding до частіших слів, мережа може правильно інтерпретувати його тон. Це пом'якшує проблему синонімів, опечаток, варіантів написання. Глибокі моделі також масштабуються: зі збільшенням даних їх якість зазвичай зростає (тим часом як деякі класичні алгоритми можуть «перенасичуватися»). За наявності великих дата-сетів глибоке навчання максимально розкриває свій потенціал, часто перевершуючи людську узгодженість (human agreement) у складних задачах тональності.

Попри успіхи, є і суттєві виклики для завдань визначення тональності текстів. Перш за все – вимоги до даних і ресурсів. Для навчання нейронної мережі потрібно значно більше розмічених даних, ніж для класичних алгоритмів, щоб модель не перенавчилася і змогла охопити різноманіття мовних конструкцій. Збирання такої кількості даних не завжди можливе, особливо для малорозвинених мов або вузьких предметних областей. По-друге, глибоке навчання є обчислювально затратним: навчання моделей займає години чи дні навіть на сучасних GPU, а розгортання в продакшн потребує значних потужностей для швидкої обробки (особливо для довгих текстів). Для багатьох прикладних сценаріїв (наприклад, аналіз мільйонів твітів у реальному часі) це може стати бар'єром – класичні методи в таких випадках швидші. По-третє, низька інтерпретованість: нейронна мережа – це "чорний ящик" зі сотнями тисяч параметрів, і зрозуміти, чому саме вона дала ту чи іншу оцінку тексту, дуже складно. Науковці намагаються досліджувати увагу (attention weights) в рекурентних моделях, візуалізувати активації, однак повної прозорості все ж немає. Це може бути критично в деяких галузях (медицина, право), де потрібно пояснення рішення алгоритму. І нарешті, нейронні моделі теж не всесильні – вони можуть помилятися на тих самих складних випадках, що й люди, або через випадкові шуми даних. Відомо, що нейронні мережі можуть вловлювати невидимі патерни, іноді хибні (наприклад, асоціювати

тональність з авторами чи джерелами даних у корпусі, якщо ті нерівномірно розподілені). Тому потрібен обережний підхід до підготовки даних і валідації результатів [7].

Незважаючи на ці труднощі, переваги глибокого навчання зробили його домінуючим підходом у аналізі тональності останніми роками. Подальший розвиток пов’язаний із ще складнішими архітектурами і переднавченими моделями, про які йдеться в наступному підрозділі.

1.5.4 Методи глибинного навчання (нейронні мережі)

Остання революція в технологіях NLP – поява трансформерних моделей (Transformer-based models), таких як BERT (Bidirectional Encoder Representations from Transformers), RoBERTa, XLNet, GPT та ін. Трансформери були запропоновані в 2017 році і здійснили переворот у багатьох задачах обробки мови, включно з аналізом тональності. Їх новизна – архітектура Transformer, що використовує механізм самоуваги (self-attention) для моделювання зв’язків між всіма словами в реченні одночасно. Це дозволяє ефективно враховувати довготривалі залежності і контекст без рекурентного проходження, як в LSTM.

В контексті аналізу тональності найзначущим є те, що з’явилися попередньо навчені трансформерні моделі мови – наприклад, модель BERT була попередньо навчена на величезному масиві текстів (Wikipedia + книги, близько 3,3 млрд слів) вирішувати завдання мовного моделювання. У результаті BERT “навчився” загальної мовної інформації: значень слів, контекстних залежностей, навіть деяких нюансів семантики і синтаксису. Цю готову модель можна потім донавчити (fine-tune) на конкретній задачі, зокрема класифікації тональності, маючи відносно невеликий розмір тренувальної вибірки. Таким чином, трансформери несуть у собі загальні знання про мову, які дуже корисні для визначення тональності. Це частково розв’язує проблему браку даних: навіть якщо корпус тональності невеликий, попередні знання BERT дозволяють досягти високої точності. Дослідження підтверджують, що трансформерні моделі значно перевершують попередні алгоритми. Наприклад, у порівняльному експерименті на твіттер-даних модель BERT після

донавчання показала найкращий результат – ~85,4% точності, обігнавши як класичні алгоритми (SVM, Naive Bayes), так і навіть LSTM-мережу.

Серед трансформерів для аналізу тональності найбільше застосовуються BERT і його модифікації. Існують багатомовні версії (mBERT) і модель XLM-RoBERTa, що охоплюють десятки мов і дозволяють одразу аналізувати тексти різними мовами, не тренуючи окрему модель під кожну (це актуально, якщо система має обробляти, наприклад, і англійські, і українські тексти). Експериментально показано, що mBERT і XLM-R можуть визначати тональність навіть для мов, яких не було у тренуванні, за рахунок спільних многомовних представлень. Також з'являються спеціалізовані трансформери, навчені на українській мові (наприклад, *Ukrainian-RoBERTa, ukr-Electra*), які встановлюють нові рекорди точності в українських NLP-задачах.

Висока якість трансформерів проявляється на різних типах даних. Наприклад, модель RoBERTa-LSTM (гібрид, що поєднує трансформер і рекурентну мережу) показала ~93% точності на IMDb та ~92% на твіттер-датасеті про авіалінії. Інша робота запропонувала BERT з додатковим CNN+BiLSTM шаром, що дало 93–97% точності на кількох наборах (відгуки авіаліній, про автівки, політичні твіти, IMDb). Сучасні експериментальні моделі, як-от ST-GCN (Sentiment Graph Convolutional Network з трансформерною увагою), досягають уже 95%+ на стандартних бенчмарках (SST-2, IMDb). Фактично, трансформерні архітектури забезпечили якісний стрибок у аналізі тональності: те, що раніше вважалося стелею (порядку 90% точності), тепер наближається до 95–97% на багатьох задачах. Для деяких специфічних датасетів повідомляється навіть про ~99% точності – наприклад, DistilBERT досяг ~99,6% на задачі класифікації тональності відгуків Amazon. Це свідчить, що на формалізованих текстах модель майже не помиляється [7].

Першочерговою перевагою такого методу є висока точність і надійність. Моделі на основі BERT встановлюють state-of-the-art результати практично на всіх відкритих тестових наборах для sentiment analysis. Вони краще розуміють контекст: на відміну від LSTM, який читає текст послідовно, трансформер аналізує всю фразу цілком, тому навіть складні граматичні конструкції (напр.,

довгі речення зі вставними фразами) опрацьовуються ефективніше. Трансформери також добре справляються з довгими текстами – вони можуть утримувати увагою зв’язки між далекими словами. Великим плюсом є перенесення знань: модель, навчена на одній мові чи домені, може бути швидко переналаштована на інший. З практичного погляду, використання попередньо навчених моделей спрощує розробку – замість навчання “з нуля” нейромережі, що потребує дуже багато даних, ми беремо готовий BERT і довчаємо його декілька епох на своєму наборі. Це значно скорочує необхідний обсяг даних і час навчання. Так, в експерименті було показано, що навіть на невеликому датасеті тональності у фінансах трансформер перевершив інші алгоритми. Крім того, є оптимізовані версії моделей: DistilBERT, ALBERT, TinyBERT – зі зменшеним числом параметрів. Вони працюють швидше, займають менше пам’яті, а точність втрачають лише незначно. Приміром, DistilBERT забезпечує ~99% точності проти 99.6% у повноцінного BERT на певному тесті, проте тренується та інферує помітно швидше. Це відкриває можливості використання трансформерів у ресурсно обмежених середовищах або для мобільних застосунків.

Головний недолік – це висока ресурсомісткість. Базова модель BERT має 110 млн параметрів, а більш просунуті – до мільярда. Їх повне навчання – надзвичайно тривалий процес, доступний лише великим «злим» корпораціям чи дослідницьким лабораторіям. Хоча fine-tuning набагато легший, все одно для обробки кожного тексту потрібні суттєві обчислення. Упровадження BERT на CPU може бути повільним, особливо для великих пакетів даних; часто потрібен GPU навіть на етапі використання, що не завжди зручно для продакшн-систем. Другий момент – обсяг пам’яті: великі моделі займають сотні мегабайт (BERT ~400 МБ), що ускладнює їх використання у вбудованих системах чи розгортання великої кількості моделей. Знову ж таки, це частково розв’язується використанням стиснутих версій (DistilBERT ~240 МБ) чи квантованих моделей. Ще один недолік – все та ж інтерпретованість. Якщо звичайну нейромережу важко інтерпретувати, то трансформер із його багатьма шарами самоуваги – ще складніший. Хоч зараз активно досліджують увагу

трансформерів, щоб зрозуміти, на які слова модель звертає увагу при ухваленні рішення (зокрема, є роботи, що вивчають чи використовує BERT словники тональності у своїх внутрішніх представленнях), але ці аналізи поки в експериментальній фазі. В прикладних застосуваннях часто доводиться сприймати трансформер як чорний ящик із гарантовано високою якістю, але низькою пояснюваністю.

Підсумовуючи, трансформерні моделі на сьогодні є найпотужнішим інструментом для аналізу тональності тексту. Вони вимагають більше ресурсів, але й дають найкращі результати. Вибір між класичними, глибокими та трансформерними методами залежить від завдань проекту: обсягу доступних даних, необхідної точності, швидкодії та доступних обчислювальних потужностей. У багатьох випадках доцільно використовувати гібридні рішення або ж рішення глибокого навчання.

Таким чином, у розрізі порівняльного аналізу можна продемонструвати, що універсально найкращого методу не існує – вибір залежить від вимог конкретного завдання. Таблиця 1.1 нижче узагальнює порівняння підходів:

Таблиця 1.1

Порівняння основних підходів до аналізу тональності тексту за ключовими критеріями.

Критерій	**Лексиконний**	**ML (класичний)**	**Глибокий**	**Трансформер**
Точність	Невисока (домінує нейромережа)	Середня/висока (до ~90%)	Висока (до ~95%)	Дуже висока (SOTA, >95% на сприятл. даних)
Швидкість (інференс)	Дуже висока (миттєво)	Висока	Середня	Низька
Обчисл. ресурси	Мінімальні (CPU)	Невеликі (CPU)	Вагомі (GPU бажано)	Дуже великі (GPU/TPU)
Необх. дані для навч.	Не потребує (словник)	Потребує (тис. прикладів)	Потребує (десятки тис.)	Fine-tuning: мало (тисячі)

Таблиця 1.1 продовження

Критерій	**Лексиконний**	**ML (класичний)**	**Глибокий**	**Трансформер**
Інтерпретованість	Відмінна (прозора)	Помірна	Низька	Дуже низька
Гнучкість до нових доменів	Вимагає оновлення словника	Перенавчання моделі	Перенавчання/довчання	Найкраща переносимість

З таблиці бачимо, що для задач, де критичною є швидкість і прозорість (наприклад, оперативний моніторинг соцмереж), варто розглядати лексиконні або прості ML-рішення – вони дадуть результат майже в реальному часі і дозволять зрозуміти, чому система робить той чи інший висновок. Якщо ж на першому місці точність і є достатньо ресурсів, безперечно варто обрати глибокі або трансформерні моделі, які забезпечують значно кращу якість класифікації. У випадку обмеженості в розмічених даних, оптимальним рішенням є transfer learning з трансформером – навіть маленький датасет можна збалансувати великими знаннями моделі, отриманими з інших джерел. Крім того, компромісним варіантом може бути використання спрощених трансформерів (DistilBERT, TinyBERT), що все ще дають високу точність, але працюють швидше.

1.5.5 Гібридні та ансамблеві підходи

Окремо слід згадати про тенденцію до поєднання методів для досягнення кращих результатів. Дослідники відзначають, що комбінація лексиконного та машинного підходів може давати синергію – зокрема, в роботі О.Залуцької та ін. (2023) показано, що об’єднання словникового аналізу з класифікатором (SVM, Logistic Regression, XGBoost) перевершує кожен з методів окремо. Лексикон вносить інтерпретованість і стійкість, а ML-класифікатор – тонше підганяє результат під специфіку даних, досягаючи точності понад 88% на україномовних текстах соцмереж. Інший напрям – ансамблі моделей машинного навчання: об’єднання кількох різних алгоритмів шляхом

голосування або усереднення їх прогнозів. Це підвищує стабільність і часто дає виграш у точності на 1-3%.

Для сучасних глибоких підходів також актуальні ансамблі: наприклад, мульти-модельні ансамблі, де об’єднуються CNN, LSTM і, скажімо, BERT. Відомо, що трансформери можна покращити, додавши до них спеціалізовані шари: комбінація BERT + CNN + BiLSTM показала кращу повноту і точність, ніж просто BERT. Очевидно, це ціною збільшення ресурсоємності. В умовах, коли якість критично важлива, такі гібриди виправдані – наприклад, для аналізу тональності в медичній сфері може використовуватися комбінована модель: словниковий модуль виявляє специфічні медичні терміни як позитивні/негативні, а нейромережа вирішує загальну тональність.

В цілому, сучасний світовий тренд – це перехід до трансформерних архітектур та великих мовних моделей для задач аналізу тональності. Проте на практиці часто застосовують і багаторівневі рішення: наприклад, система спочатку фільтрує очевидно позитивні/негативні випадки простим словниковим методом, а більш нейтральні чи складні випадки передає на обробку потужній нейромережі. Така каскадна схема дозволяє зменшити середній час обробки, зберігаючи високу якість там, де це необхідно.

Отже, провівши порівняння, можемо зробити висновок, що кожен з підходів має свою нішу. В контексті даної дисертації, де метою є розробка ефективного рішення для визначення тональності, доцільно врахувати ці результати: прагнучи високої точності, ми будемо орієнтуватися на сучасні нейромережеві моделі, але також шукатимемо способи підвищення їх ефективності (швидкодії, меншої залежності від великих даних) – можливо, через поєднання зі спрощеними методами або оптимізацію архітектури.

1.6 Формалізація задачі дослідження та постановка часткових задач

Метою даного дослідження є розробка методу та програмної компоненти для автоматичного визначення тональності текстових повідомлень на основі технологій обробки природної мови (NLP). Для досягнення поставленої мети необхідно вирішити наступні часткові завдання:

1. Провести аналіз особливостей інформаційного контенту інтернет-медіа, визначити ключові характеристики та специфіку новинних текстів.

2. Здійснити аналіз існуючих методологічних підходів до визначення тональності текстових повідомлень за технологіями NLP, оцінити їх переваги та недоліки.

3. Виконати аналіз існуючих технологічних рішень, що використовуються для визначення тональності текстових повідомлень, визначити їхні можливості, обмеження та перспективи застосування.

4. Розробити власний метод визначення тональності текстових повідомлень, що враховує специфіку українськомовних новинних ресурсів та дозволяє ефективно класифікувати тональність повідомлень.

5. Реалізувати програмну компоненту, яка інтегрує розроблений метод у вигляді практично придатного програмного забезпечення, приділивши увагу продуктивності, точності та можливості масштабування рішення.

Математична постановка задачі:

Нехай задано множину текстових повідомлень $T = \{t_1, t_2, ..., t_n\}$, де кожне повідомлення t_i являє собою текстову послідовність слів: $t_i = (w_1, w_2, \ldots, w_k)$, w_j – j-е слово повідомлення. Кожне повідомлення t_i має бути класифіковане за ознакою тональності у множині класів $C = \{c_1, c_2, ..., c_m\}$, де c_1 – позитивна тональність, c_2 – нейтральна тональність, c_3 – негативна тональність (у випадку двокласової класифікації множина скорочується до $\{c_1, c_3\}$).

Задача визначення тональності тексту формалізується як задача класифікації:

$F: T \rightarrow C$,

де F – класифікаційна модель (в нашому випадку – нейромережева модель на базі архітектури Embedding + LSTM), що здійснює перетворення текстового повідомлення у відповідну мітку класу тональності.

Формально, задача навчання полягає в оптимізації параметрів нейронної мережі θ шляхом мінімізації функції втрат:

$$L(\theta) = -\sum_{n=1}^{N}\sum_{m=1}^{M} y_{nm}\log(p_{nm})$$

де:

- N – кількість текстових повідомлень у тренувальній вибірці;
- M – кількість класів тональності;
- y_{nm} – індикаторний показник, що дорівнює 1, якщо текст n належить класу m, і 0 – у протилежному випадку;
- p_{nm} – ймовірність належності тексту n класу m, отримана в результаті роботи нейромережевої моделі.

Реалізація програмної компоненти включає:

1. Модуль збору текстових повідомлень з новинних RSS-стрічок.
2. Модуль попередньої обробки текстів (лематизація, токенізація, очищення текстів).
3. Модуль нейромережевої моделі класифікації (LSTM, Embedding-матриця).
4. Модуль візуалізації та аналітики результатів (графіки динаміки тональності, матриця помилок).

Таким чином, виконання зазначених завдань забезпечить створення та впровадження ефективного методу та програмної компоненти визначення тональності текстових повідомлень для українськомовних інтернет-медіа, що підтвердить практичну корисність та перспективність запропонованого підходу.

ВИСНОВОК ДО РОЗДІЛУ 1

У першому розділі проведено огляд предметної області аналізу тональності текстів (sentiment analysis) та сучасних NLP-технологій для її реалізації. Розглянуто специфіку контенту інтернет-медіа: великий обсяг даних, швидка генерація і поширення, емоційно забарвлена подача інформації та наявність неформальної лексики. Виявлено, що ці фактори зумовлюють потребу в автоматизованому аналізі настроїв, оскільки тональність онлайн-контенту впливає на суспільну думку і її відстеження є важливим завданням.

Огляд існуючих технологічних рішень показав, що доступні як комерційні сервіси (Google, IBM, Azure тощо) із підтримкою багатьох мов та високим рівнем аналізу (визначення емоцій), так і open-source інструменти (VADER, TextBlob, трансформерні моделі), які можна використати для розробки власної системи. Методологічно виділено три основні підходи: словниковий, машинного навчання та гібридний. Словниковий підхід простий у реалізації, але обмежений якістю словника; ML-підходи забезпечують вищу точність за наявності даних для навчання; глибокі нейронні мережі дають найкращі результати на складних задачах, хоча потребують більше ресурсів. Відзначено, що для нашої задачі, з урахуванням вимог до швидкодії та обмеження ресурсів, доцільно розглянути оптимізовані підходи машинного навчання або їх поєднання зі словниковими методами – такі рішення можуть бути достатньо точними, як показують сучасні дослідження, і водночас легкими в розгортанні.

На основі аналізу сформульовано мету і завдання дослідження. Далі, у розділі 2, буде представлено розроблений метод визначення тональності та його програмну реалізацію, що відповідають визначеним вимогам і враховують висновки огляду.

РОЗДІЛ 2

РОЗРОБКА ПРОГРАМНОЇ КОМПОНЕНТИ ВИЗНАЧЕННЯ ТОНАЛЬНОСТІ ТЕКСТОВИХ ПОВІДОМЛЕНЬ ЗА ТЕХНОЛОГІЯМИ NLP

2.1. Розробка методу визначення тональності текстових повідомлень

Методи машинного навчання, зокрема із застосуванням глибоких нейронних мереж, демонструють вищу точність завдяки здатності навчатися на великих обсягах даних. Такі моделі автоматично виявляють складні залежності між словами та тональністю тексту. Лексиконні (словникові) методи прості у реалізації, проте часто не враховують контекстуального значення слів та не справляються зі складними мовними конструкціями (наприклад, іронією чи запереченнями).

Як я зазначав у першому розділі, для англомовного контенту існують великі розмічені корпуси (набори даних) та готові моделі для визначення тональності. Натомість для української мови обсяг доступних даних обмежений, а готові рішення з’явилися відносно нещодавно, та не знаходяться у відкритому доступі. Пряме застосування англомовних моделей на українських текстах є неефективним через мовні відмінності. Можливі шляхи вирішення цієї проблеми включають:

- Створення власного датасету: ручне збирання та розмітка українських текстів за тональністю. Це забезпечує якісні дані, проте є трудомістким і довготривалим процесом.
- Переклад наявних корпусів з інших мов: існуючі англомовні або російськомовні набори даних можна перекласти українською за допомогою автоматичних перекладачів. Такий підхід швидкий і дозволяє отримати значний обсяг тренувальних даних, однак потребує перевірки якості перекладу та збереження тональності під час перекладу.
- Використання багатомовних моделей NLP: сучасні моделі на кшталт mBERT, XLM-RoBERTa тощо здатні розуміти тексти різними мовами, зокрема українською. Їх можна донавчати (fine-tune) на українських даних. Цей варіант забезпечує високу точність, проте вимагає значних обчислювальних

ресурсів і глибокого розуміння архітектури моделей.

Беручи до уваги обмежені часові та ресурсні рамки проєкту, а також бажання продемонструвати весь цикл розробки (від збору даних до створення працюючого застосунку), було обрано комбінований підхід. Запропонований метод базується на глибокому навчанні (нейронній мережі для класифікації тексту) у поєднанні з автоматичним перекладом корпусу даних для розширення навчальної вибірки. Новизна підходу полягає в адаптації існуючих технологій до україномовного контенту шляхом поєднання власноруч зібраного українського корпусу повідомлень із автоматично перекладеним англомовним корпусом; навчання окремих моделей на кожному з корпусів та порівняння їх характеристик, а також експериментів з об'єднанням даних; розробки універсальної програмної компоненти, яка підтримує кілька мовних корпусів і режимів роботи (бінарна класифікація "позитив/негатив" та багатокласова "позитив/нейтральний/негатив").

Спочатку було сформовано невеликий український корпус текстів із тональними мітками. З огляду на те, що в українському сегменті інтернету відсутній великий відкритий набір даних для аналізу тональності, довелося створити його самостійно. Для цього вибиралися тексти з соціальних мереж і новинних сайтів, які явно виражають позитивне, негативне або нейтральне ставлення. Кожне повідомлення отримало мітку тональності: *"позитивне"*, *"нейтральне"* або *"негативне"*. Усього український корпус склав декілька сотень повідомлень (після очищення та балансування за класами придатними до використання було близько 100–200 прикладів кожної категорії).

Для збільшення обсягу даних було використано перекладений Twitter-корпус. Було обрано наявний англомовний набір даних із Twitter-повідомлень (твітів) з тональними мітками. Зокрема, було зібрано ~5 000 твітів англійською мовою, які містили як позитивні, так і негативні висловлювання (нейтральні відсутні, щоб зосередитися на бінарній класифікації "позитив/негатив"). Цей корпус було автоматично перекладено українською мовою за допомогою сервісу машинного перекладу, що зайняло близько 3 годин. При перекладі особлива увага приділялася тому, щоб зберегти полярність висловлювань: наприклад, негативні повідомлення після перекладу мали залишатися

негативними за змістом. Отриманий перекладений корпус твітів було використано для навчання окремої моделі, орієнтованої на визначення тільки позитивної чи негативної тональності.

Таким чином, запропонований метод включає дві стратегічні лінії:

1. Модель для українського корпусу (3-класова класифікація): навчена на автентичних українських даних, здатна розрізняти позитивний, нейтральний і негативний тон. Ця модель враховує особливості української мови стилістично та лексично, проте обмежена невеликим обсягом тренувальних даних.

2. Модель для перекладеного корпусу (2-класова класифікація): навчена на значно більшому обсязі даних (перекладених твітів), але лише для визначення полярності (позитив vs. негатив). Вона покликана продемонструвати, якої точності можна досягти, використовуючи великий датасет, хоч і отриманий непрямим шляхом. Цей підхід базується на припущенні, що автоматичний переклад достатньо якісно передає тональність, і тому модель може навчитися розпізнавати позитивні та негативні настрої українською мовою.

Для класифікації текстів за тональністю було обрано нейромережевий підхід. Модель являє собою багатошарову нейронну мережу, специфічно архітектуровану для обробки послідовностей слів. Зокрема, використано шар вбудовування слів (Embedding), який перетворює кожне слово в тексті на багатовимірний числовий вектор. Ці вектори навчуються таким чином, що слова з подібним значенням мають близькі представлення. Далі йде рекурентний шар типу LSTM (Long Short-Term Memory – довга короткострокова пам’ять), який здатний враховувати контекст слова, використовуючи інформацію про попередні слова в реченні. LSTM було обрано через його ефективність у задачах аналізу послідовностей і здатність зберігати довгострокові залежності в тексті (наприклад, пам’ятати наявність заперечення “не” декількома словами раніше).

Математично архітектуру LSTM-шару можна описати наступним чином: LSTM-нейрон використовує три основні вентилі (гейти):

1. Вхідний вентиль (input gate):

$$i_t = \sigma(W_i * [h_{t-1}, x_t]\ b_i$$

2. Вентиль забування (forget gate):

$$f_t = \sigma(W_f * [h_{t-1}, x_t]\ b_f$$

3. Вихідний вентиль (output gate):

$$o_t = \sigma(W_o * [h_{t-1}, x_t]\ b_o$$

Далі кандидатний стан клітини (cell state) оновлюється наступним чином:

$$\hat{\mathrm{C}}_t = \tanh(W_c * [h_{t-1}, x_t]\ b_c)$$

Стан клітини оновлюється з урахуванням попереднього стану C_{t-1} і визначається вентилями забування та вхідним вентилем:

$$\mathrm{C}_t = f_t \odot C_{t-1} + i_t \odot \hat{\mathrm{C}}_t$$

Вихідний вектор прихованого стану h_t формується на основі стану клітини через вихідний вентиль:

$$h_t = o_t \odot \tanh(\mathrm{C}_t)$$

де:

- x_t – вхідний вектор на момент часу ttt (наприклад, вектор слова після embedding-шару);
- h_{t-1} – вихід попереднього кроку (прихований стан);
- W_i , W_f , W_o , W_c – матриці вагових коефіцієнтів для відповідних вентилів;
- $b_i, b_f, b_o, b_c,$ – вектори зміщення;
- $\sigma\ (*)$ – сигмоїдна функція активації;
- $\tanh(*)$– гіперболічний тангенс;
- $\odot$ – поелементне множення (операція Адамара).

Таким чином, архітектура LSTM дозволяє ефективно зберігати інформацію та управляти потоком контекстної інформації при обробці текстових послідовностей, що робить її оптимальним вибором для аналізу тональності тексту.

Поверх LSTM додано один або кілька щільних (Dense) шарів для прийняття рішення, а вихідний шар — це шар Softmax, який видає ймовірності належності тексту до кожної з передбачених категорій тональності. Для

бінарної моделі використовується 2-вихідний Softmax (або еквівалентно сигмоїдна активація на одному виході), а для 3-класової — 3-вихідний Softmax.

Навчання моделей здійснювалося з функцією втрат categorical crossentropy (крос-ентропія) та оптимізатором Adam, який добре зарекомендував себе в задачах текстової класифікації завдяки швидкій збіжності. Навчальні дані було розділено на тренувальну і тестову вибірки: для українського корпусу через його малий обсяг використовувалася стратегія k-кратної перевірки (k-fold cross-validation) або утримано окремо 20% для тестування; для великого перекладеного корпусу 20% твітів відкладено під тест під час навчання. Модель тренувалася ітеративно (епохами) до збіжності метрики точності на валідаційній вибірці. Щоб запобігти перенавчанню, використовували Early Stopping — моніторинг втрати на валідальних даних та зупинка навчання, коли покращення перестає спостерігатися.

Новизна методу полягає в тому, що він поєднує переваги двох світів — якісні, але малі українські дані, і великі, але іншомовні дані, адаптовані через переклад. На відміну від прямого використання багатомовної моделі “чорного ящика”, такий підхід дозволяє гнучко будувати та вдосконалювати власну архітектуру, зрозуміти її роботу і точно контролювати склад корпусу. Комбінація підходів проявляється і в тому, що розроблена програмна компонента здатна працювати з кількома моделями: можна перемикатися між 2-класовою високоточною моделлю (твіти) та 3-класовою моделлю (українські тексти з нейтральним класом). Таким чином, продемонстровано спосіб вирішення задачі аналізу тональності для мови з обмеженими ресурсами: шляхом машинного перекладу даних і побудови спеціалізованої моделі. Цей метод потенційно може бути розширений на інші типи текстів (наприклад, новини, відгуки) і на інші мови, що знаходяться в схожій ситуації з доступністю даних. Нижче наведено структурну схему методу (рис. 2.1.)

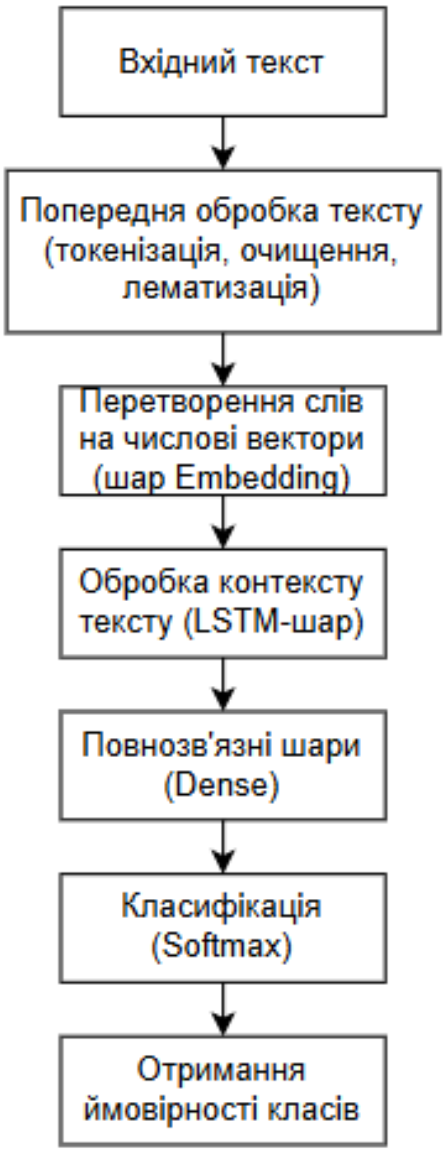

Рис. 2.1. Структурна схема розробленого методу

2.2. Розробка програмної компоненти визначення тональності (архітектура)

Після визначення методології постає завдання реалізувати її у вигляді програмного забезпечення. Ця секція присвячена опису процесу розробки програмної компоненти: від аналізу вимог до системи — до архітектурного проєктування та безпосередньо програмної реалізації. Розробка виконувалася ітеративно з перевіркою на практиці кожного із етапів.

2.2.1 Інженерія вимог

На початковому етапі розробки було здійснено збір і аналіз вимог до програмної компоненти аналізу тональності. Інженерія вимог включала визначення функціональності системи, характеристик продуктивності,

інтерфейсу користувача та інших важливих аспектів. Основні вимоги (функціональні та нефункціональні) до системи були сформульовані наступним чином.

Система повинна автоматично визначати тональність довільного текстового повідомлення українською мовою. Вихід — тональна мітка: *позитивна*, *нейтральна* або *негативна* (у разі використання 3-класової моделі) або *позитивна/негативна* (для 2-класової моделі). Програмна компонента має підтримувати принаймні два режими або моделі: (1) модель, навчена на повному українському корпусі з нейтральною тональністю, (2) модель, навчена на перекладеному Twitter-корпусі (бінарна класифікація). Користувач (або система) повинні мати змогу вибрати потрібний режим аналізу. Передбачено наявність простого у користуванні графічного інтерфейсу (GUI), що дозволить завантажувати джерела даних (наприклад, останні новини) та оперативно отримувати результат. Інтерфейс має бути локалізований українською мовою і не вимагати від користувача технічних знань. Система повинна не лише видавати текстові результати (мітки тональності), але й забезпечувати наочне відображення аналітики — наприклад, графіків розподілу тональностей для набору повідомлень, матриці помилок для оцінки якості моделі тощо. Це допоможе користувачу або аналітику краще зрозуміти результати роботи моделі.

Модель тональності має демонструвати високу точність класифікації на тестових даних (бажано не менше 85–90% для основних класів). Система повинна працювати у реальному часі або близькому до нього: аналіз одного повідомлення має тривати частки секунди, що дозволить інтерактивно отримувати результати. При обробці пакетів (наприклад, списку новин) час відповіді також має бути прийнятним (кілька секунд для десятків повідомлень). Архітектура програмної компоненти повинна бути модульною, щоб за потреби можна було замінити модель (наприклад, інтегрувати більш сучасну або багатомовну) без повної переробки системи. Слід передбачити можливість додавання нових джерел даних або розширення набору підтримуваних мов. Кодова база повинна бути структурована і

задокументована. Система має запускатися стандартними засобами (наприклад, через команду запуску веб-інтерфейсу) та мати зрозумілі налаштування. Бажано використовувати поширені мови програмування та бібліотеки (Python, бібліотеки NLP), щоб забезпечити легкість підтримки проєкту спільнотою або іншими розробниками.

Функціонал компоненти має полягати в тому, що компонента здійснює парсинг заголовків новин із визначених українських веб-ресурсів через RSS-стрічки, передбачає токенізацію, очищення від зайвих символів та лематизацію українських текстів, що забезпечує високу точність подальшого аналізу. Саме визначення тональності функціонально має бути реалізовано на базі програми, яка здатна класифікувати текстові повідомлення на «позитивні», «негативні» та «нейтральні», або на двокласову шкалу («позитивні» і «негативні») залежно від вибору моделі користувачем. Компонент надає можливість візуалізувати матрицю помилок, а також отримати детальний класифікаційний звіт, що дозволяє користувачу аналізувати й оцінювати ефективність роботи моделей, а також формує інтерактивні графіки розподілу тональностей для кожного джерела новин, що дозволяє порівнювати медіаресурси за тональністю представленої інформації. Користувач має легко перемикатися між моделями, натренованими на різних наборах даних (український корпус та перекладений Twitter-корпус), залежно від конкретної задачі аналізу.

Сформовані вимоги були узгоджені з науковим керівником та потенційними користувачами. На їх основі розпочато проектування системи, зосереджене на тому, щоб реалізація відповідала цим вимогам (зокрема, обрано технологію Streamlit для швидкого створення UI, закладено можливість перемикання між моделями тощо).

2.2.2 Архітектурне проєктування програмної компоненти

Архітектура програмної компоненти визначення тональності спроєктована таким чином, щоб розділити систему на логічні модулі, кожен з яких відповідає за свою підзадачу. Такий підхід забезпечує розділення відповідальностей між частинами системи, високу масштабованість і зручність супроводу. Було розроблено багаторівневу архітектуру, що включає рівень

підготовки та зберігання даних, рівень моделі машинного навчання, та рівень презентації результатів користувачу. Нижче наведено структурну схему програмної компоненти (рис. 2.2.) та блок-схему її алгоритму (рис. 2.3).

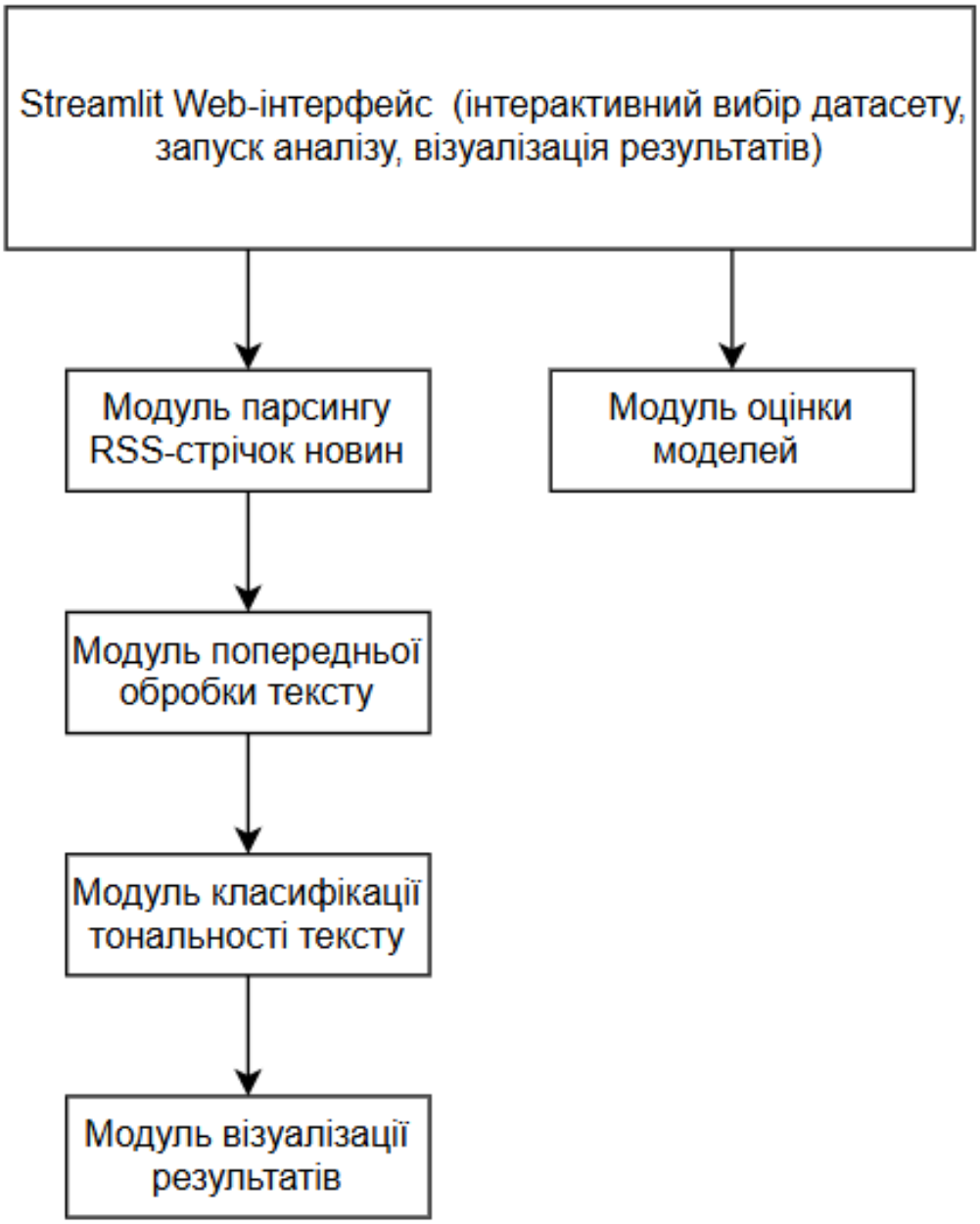

Рис. 2.2. Узагальнена структурна схема програмної компоненти

Рис. 2.3. Блок схема алгоритму програмної компоненти

На рівні даних передбачено модулі для завантаження і підготовки датасетів. Він відповідає за підготовку та обробку текстових даних. Він включає два незалежні модулі. Оскільки використовується два різні корпуси (український та перекладений), архітектурно це реалізовано через окремі

компоненти: модуль dataset_ua для роботи з українським корпусом і модуль dataset_translated для роботи з перекладеним твіттер-корпусом. Кожен з цих модулів містить функції для зчитування даних (наприклад, з CSV-файлу), очищення тексту (видалення зайвих символів, URL, хештегів, якщо є, тощо), токенізації тексту (перетворення слів у числові індекси) та розбиття на тренувальну і тестову вибірки. На цьому ж рівні визначені параметри підготовки даних — такі як MAX_WORDS (максимальний розмір словника, тобто скільки найбільш частотних слів враховувати) та MAX_LEN (максимальна довжина послідовності слів у повідомленні, до якої здійснюється паддінг або усічення). Дані, підготовлені на цьому рівні, передаються на вхід моделі у вигляді числових тензорів (матриць).

На рівні логіки спроєктовано компонент, що безпосередньо виконує класифікацію тональності та реалізує основну функціональність проєкту. Архітектура моделі містить шари Embedding, LSTM та Dense. Для гнучкості розробки вирішено не “зашивати” архітектуру жорстко в код одного файлу, а дозволити використання різних моделей. Тому архітектура оформлена у вигляді функції або класу, який можна викликати з різними параметрами. В даному проєкті архітектура для обох моделей подібна, відрізняється лише вихідний шар (2 або 3 нейрони) та деякі гіперпараметри (наприклад, розмір словника чи довжина послідовності можуть різнитися, виходячи з характеру даних твітів і українських текстів). Модель була реалізована із застосуванням бібліотеки TensorFlow (Keras), що полегшує визначення шарів та навчання моделі. Навчені моделі зберігаються на диску у файлах формату .h5 (формат збереження моделей Keras), наприклад, model_ua.h5 для української моделі та model_translated.h5 для моделі на перекладених твітах. Це дозволяє легко завантажувати потрібну модель під час ро

На рівні застосунку та представлення спроектовано компоненту, що відповідає за взаємодію з користувачем, відображення результатів та високорівневе керування процесом. Було обрано архітектуру веб-застосунку: інтерфейс реалізовано засобами Streamlit, що дозволяє створювати веб-сторінку для взаємодії в режимі реального часу з Python-кодом у бекенді. У

архітектурі це виокремлено як модуль streamlit_app, який містить увесь код, пов’язаний з UI, включаючи виклики до моделей та обробку дій користувача (натискань кнопок, введення тексту тощо). Цей модуль імпортує функції з рівня даних та моделі, та координує їх роботу в залежності від дій користувача.

В цілому, архітектуру можна уявити як конвеєр обробки даних:

1. Введення/отримання текстів: Користувач через інтерфейс або зовнішнє джерело (наприклад, парсер новин) передає один або кілька текстів для аналізу.
2. Попередня обробка: Тексти надходять у відповідний модуль підготовки даних (dataset_ua чи dataset_translated), де виконуються очищення та токенізація. Якщо використовується заздалегідь навчена модель, то слова перетворюються в індекси згідно зі словником, отриманим під час навчання (використовується об’єкт токенізатора, що був збережений або визначений наперед).
3. Класифікація моделлю: Підготовлені тензори передаються до моделі (обраної моделі тональності). Модель обчислює ймовірності належності кожного тексту до можливих класів та видає прогнозовані тональні мітки.
4. Постобробка та візуалізація: Результати класифікації повертаються на рівень застосунку. Тут вони можуть бути відображені у зручному для користувача вигляді: текстові мітки (“Тональність: негативна”) поруч з кожним повідомленням, зведені метрики (точність, повнота) у вигляді таблиці, матриця помилок у вигляді графіка, розподіл позитив/нейтраль/негатив у наборі текстів – у вигляді стовпчикової діаграми тощо.
5. Взаємодія з користувачем: Користувач, побачивши результати, може внести нові дані для аналізу, перемкнути режим моделі (наприклад, на інший корпус) або виконати інші дії (наприклад, оновити дані новин). Інтерфейс у Streamlit постійно чекає на такі дії і динамічно оновлює відображення.

Для проєктування архітектури було створено діаграми високого рівня, що відображають взаємозв’язок модулів. Зокрема, діаграма компонент відобразила три основні компоненти: *Data Loader* (набори даних), *Sentiment Model* (нейронна мережа) та *User Interface* (Streamlit-застосунок), а також потоки даних між ними. На діаграмі послідовності було показано, як при натисканні користувачем кнопки “Аналізувати” запускається послідовність

викликів: UI -> завантаження та обробка даних -> застосування моделі -> повернення результатів -> відображення на UI. Такі проектні матеріали допомогли впевнитися, що всі необхідні взаємодії враховані до початку написання коду.

Варто зазначити, що при проектуванні архітектури враховувалася можливість розвитку проєкту. Наприклад, щоб у майбутньому інтегрувати більш досконалу модель (скажімо, мультимовний трансформер), достатньо реалізувати новий модуль моделі з тим самим інтерфейсом взаємодії (функціями завантаження, прогнозування тощо) та підключити його до існуючого інтерфейсу. Така гнучкість досягається завдяки модульності: чітко визначені межі відповідальності кожної компоненти спрощують заміну або модифікацію окремих частин системи.

2.3. Програмна реалізація

На етапі програмної реалізації проєкту було написано вихідний код згідно з розробленою архітектурою. Розробку виконано мовою Python 3.10 (інтерпретована, високорівнева мова, широко застосовувана в сфері NLP завдяки багатій екосистемі бібліотек). Середовище розробки – PyCharm (проект Tonality_Analysis). Варто розглянути ключові модулі, класи та скрипти проекту, навести приклади реалізації та розповімо про труднощі, що виникали під час кодування.

Проєкт упаковано у вигляді директорії Tonality_Analysis, що містить кілька Python-модулів та допоміжних файлів. Головний скрипт запуску веб-застосунку на Streamlit – це streamlir_app.py. У цьому файлі описано структуру інтерфейсу (розміщення кнопок, випадаючих списків, заголовків, графіків), а також логіка обробки подій. Зокрема, при запуску Streamlit виконується ініціалізація (завантажуються необхідні моделі, встановлюються параметри), а далі скрипт очікує на дії користувача. Передбачено кнопки: “Парсити новини” (для отримання актуальних текстів новин із заданих джерел), “Оцінка поточної моделі” з підкнопкою “Показати точність моделі” (для відображення метрик якості обраної моделі на тестових даних) тощо. У цьому ж модулі реалізовано вибір корпусу: випадаючий список дозволяє обрати ua Український корпус або

Перекладений Twitter-корпус. В залежності від вибору, змінюються параметри датасету і завантажується відповідна модель.

Для початку розроблено модуль що отримуватиме (парситиме) тексти зі стрічки новин сайтів. В нашому проекті була реалізована можливість завантажувати останні новини з кількох українських сайтів. Модуль призначений для автоматичного збору заголовків новин та додаткових метаданих із визначених RSS-стрічок українських сайтів. Він використовує бібліотеку feedparser, яка дозволяє швидко і зручно завантажувати та обробляти інформацію з RSS-каналів у вигляді структурованих даних. Основна функція цього модуля, parse_rss_feed, приймає як вхідний параметр URL-адресу RSS-стрічки, після чого отримує її вміст за допомогою feedparser.parse та зберігає отриману інформацію у внутрішньому форматі для подальшого використання. Парсер послідовно обробляє кожен запис (entry), що міститься в завантаженому RSS-каналі, витягаючи з нього заголовок новини (поле title), URL-посилання на повний текст повідомлення (посилання береться перше зі списку доступних links), набір тегів, якщо вони наявні (отримується через поле tags), та дату публікації новини, яку форматують у стандартизований формат YYYY-MM-DD, використовуючи модуль time. У результаті формується список словників, кожен з яких описує окремий запис новини.

За модуль завантаження та підготовки українського корпусу відповідає dataset_ua.py. Він містить метод load_data() для читання CSV-файлу українського набору даних. Ця функція виконує очищення тексту (видалення небажаних символів, приведення до нижнього регістру), створює або завантажує токенізатор (Tokenizer) для перетворення слів на індекси, перетворює тексти на послідовності фіксованої довжини (pad_sequences). Також здійснюється розподіл на тренувальну і тестову множини (в заданій пропорції, наприклад 80/20). Повертає масиви X_train, X_test, y_train, y_test та об’єкт tokenizer. Зауважимо, що український корпус містить 3 класи, тому мітки у будуть представлені у вигляді категорій (наприклад, рядків “positive”, “neutral”, “negative” або закодовані як 0/1/2). Аналогічний модуль та метод існує для другого корпусу даних – dataset_translated.py. Оскільки оригінальний

англомовний датасет містив лише дві категорії тональності, в коді передбачено фільтрацію: рядки, де Sentiment не належить до списку ['Positive', 'Negative'], відкидаються, щоб модель навчалася тільки на двох класах. Це можна побачити в коді: df = df[df['Sentiment'].isin(['Positive', 'Negative'])]. Після цього аналогічно виконується токенізація текстів і підготовка тренувального та тестового наборів.

Також реалізовано модуль оцінки моделі. Він містить функцію evaluate_model(dataset) яка завантажує дані (викликаючи відповідний load_data для обраного датасету), завантажує збережену модель з файлу .h5 (наприклад, використовуючи tensorflow.keras.models.load_model('model_translated.h5')), та обчислює метрики точності на тестовій вибірці. Результатом роботи цієї функції є, зокрема, матриця помилок (confusion matrix), що зіставляє передбачені та реальні мітки. Побудова матриці реалізована з використанням бібліотеки scikit-learn (функція confusion_matrix) або через підрахунок вручну, а для її візуалізації використано matplotlib. Також обчислюється сумарна точність (частка правильно класифікованих повідомлень) та, за потреби, інші метрики (точність (precision) та повнота (recall) для кожного класу). Цей модуль дозволяє швидко отримати уявлення про якість моделі і використовується в інтерфейсі при натисканні відповідної кнопки.

Після впровадження всіх компонент і відлагодження їх взаємодії, отримали повнофункціональну програмну компоненту, що відповідає поставленим вимогам. Далі, у наступному розділі, будуть розглянуті прикладні аспекти використання цієї компоненти, зокрема, продемонстровано її роботу на практичних прикладах та наведено інструкції з експлуатації.

ВИСНОВОК ДО РОЗДІЛУ 2

У другому розділі описано процес розробки методу та програмної компоненти для визначення тональності текстових повідомлень на основі технологій NLP. Спочатку проаналізовано проблему обмеженості українських тональних даних та обґрунтовано вибір комбінованого підходу, що поєднує машинний переклад і глибоке навчання. Запропоновано новий метод, який

дозволяє значно розширити тренувальний корпус шляхом перекладу англомовних даних, зберігаючи при цьому врахування специфіки української мови через власний український підкорпус.

На основі визначених вимог побудовано архітектуру системи, яка складається з модулів підготовки даних, нейронної моделі та інтерфейсу користувача. Архітектура забезпечує гнучкість (можливість переключення між моделями, розширення функціоналу) і відповідає поставленим функціональним та нефункціональним вимогам.

Реалізовано вихідний програмний код компоненти: створено всі необхідні модулі (dataset_ua, dataset_translated, evaluate, streamlit_app та ін.), а також додаткові утиліти для парсингу зовнішніх даних. В процесі розробки вирішено ряд технічних проблем, пов’язаних із сумісністю форматів даних, інтеграцією бібліотек та оптимізацією продуктивності. Отримано дві навчені моделі тональності, збережені для подальшого використання.

Результатом розділу є повністю підготовлена до використання програмна компонента аналізу тональності текстів, яка надалі буде використана для демонстрації практичних можливостей (розділ 3) та розгляду перспектив комерційного застосування у вигляді стартап-проекту (розділ 4).

РОЗДІЛ 3

ПРИКЛАДНІ АСПЕКТИ ЗАСТОСУВАННЯ ПРОГРАМНОЇ КОМПОНЕНТИ ВИЗНАЧЕННЯ ТОНАЛЬНОСТІ ТЕКСТОВИХ ПОВІДОМЛЕНЬ ЗА ТЕХНОЛОГІЯМИ NLP

3.1. Демонстрація практичних можливостей розробки

У цьому підрозділі представлено практичні результати роботи розробленої програмної компоненти. Демонструються можливості інтерактивного графічного інтерфейсу користувача, наведено приклади класифікації реальних текстових даних та оцінено точність моделей на тестових наборах. Метою є показати, що створена система не лише теоретично працездатна, а й ефективно виконує поставлені завдання на практиці.

Інтерактивний інтерфейс (GUI) розроблений на основі Streamlit дозволяє користувачу виконати аналіз тональності безпосередньо у веб-браузері. Інтерфейс складається з декількох секцій:

- Панель налаштувань (sidebar): У лівій частині вікна знаходиться панель, де користувач може обрати режим роботи. Зокрема, випадаючий список “Оберіть корпус для аналізу” дозволяє перемикатися між двома доступними моделями: ua Український корпус (трьохкласова модель, створена власноруч) та Перекладений Twitter-корпус (двохкласова модель) (рис. 3.1).

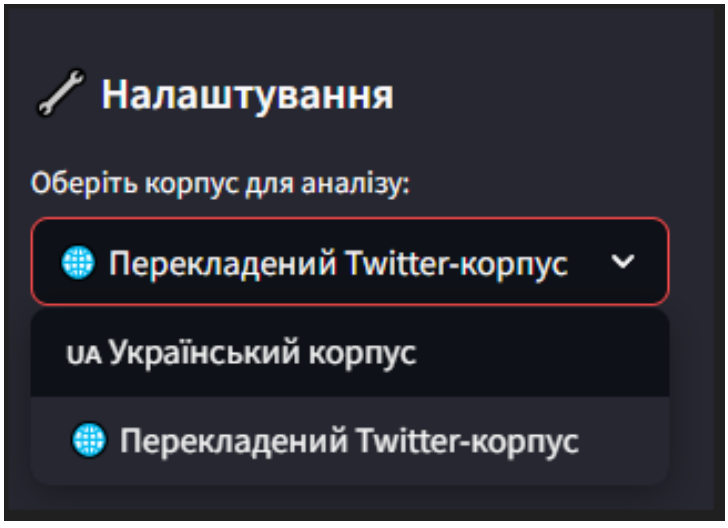

Рис. 3.1. Панель налаштувань моделі

Вибір визначає, яку модель буде використано для аналізу введених даних. Нижче в режимі реального часу відображується назва обраної натренованої попередньо моделі (рис. 3.2).

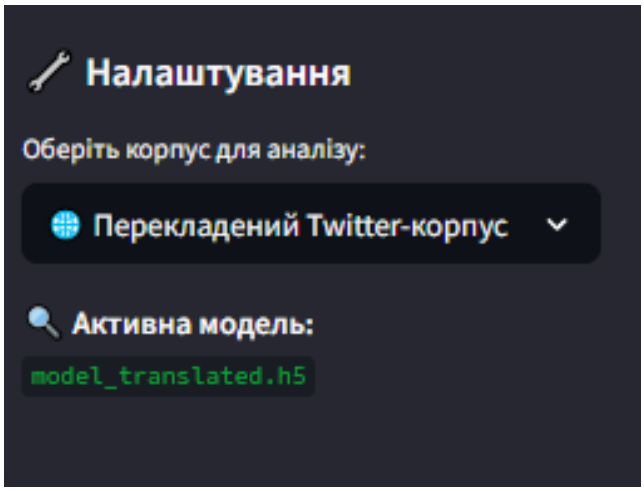

Рис. 3.2. Відображення активної моделі

- Основна область – аналіз текстів. Фактично створений інтерфейс має дві кнопки, перша з яких «Парсинг новин» відповідає за збір на 5 обраних сайтах новин (Українська правда, Економічна правда, Уніан, Укрінформ та ТСН), а друга – використовується для відображення матриці помилок обраної моделі. Результати відображаються двома способами:

1. Список новин із вказаною тональністю: Кожен заголовок новини виводиться як окремий пункт списку, поруч з ним стрілочкою позначено визначену тональність. Наприклад, після парсингу могли з’явитися такі елементи:
   - “Внаслідок атак БПЛА у Києві постраждав будинок ...” → негативний
   - “Новий парк відкрили у центрі міста” → позитивний
   - “Уряд не планує підвищувати податки наступного року” → нейтральний

   Кожен пункт містить короткий заголовок та результат класифікації. Таким чином, користувач одразу бачить, яку тональність, на думку моделі, несе кожна новина зі стрічки.
2. Зведена діаграма тональностей по джерелах: додатково система будує графік, що показує розподіл позитивних, нейтральних та негативних новин для кожного проаналізованого ресурсу. На рис. 3.3 наведено приклад такого графіку, де по осі абсцис відкладено назви сайтів (Українська правда, Економічна правда, УНІАН, Укрінформ, ТСН), а по осі ординат – кількість новин кожної тональності, знайдених на відповідному ресурсі.

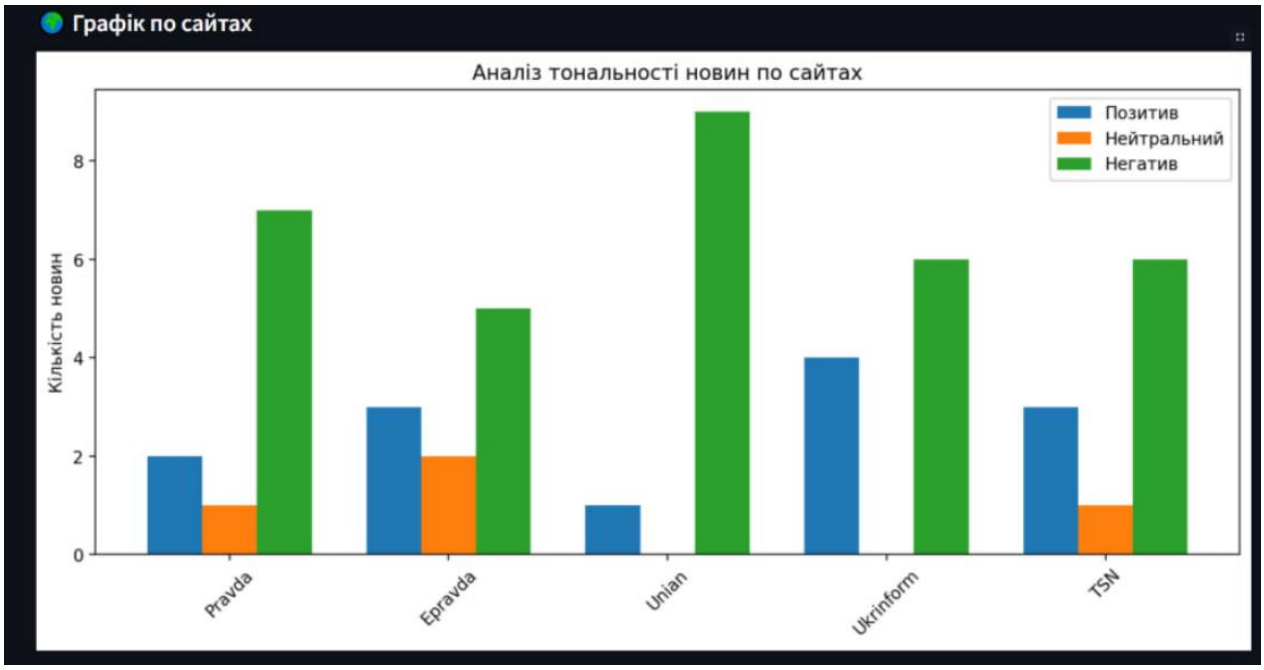

Рис. 3.3. Діаграми тональності на сайтах новин

Варто зазначити, що оскільки програма постійно бере нові новини, результат такого графіку може різнитися від абсолютно «Зради» до помірної «Перемоги».

Графік показує кількість позитивних (синій), нейтральних (помаранчевий) та негативних (зелений) новин, отриманих із різних інформаційних ресурсів. Видно, що в проаналізованому наборі переважають негативні новини (зелена колонка) на всіх представлених сайтах, що очікувано для нашого сьогодення. Позитивні новини (синя колонка) зустрічаються рідше і присутні лише на ресурсах у меншій кількості, тоді як нейтральні повідомлення (помаранчева колонка) зовсім мало або будуть взагалі відсутні якщо обирати другу модель, для якої не було розмітки з тональністю «нейтрально». Це може свідчити про те, що більшість актуальних новин носить емоційно забарвлений характер (переважно негативний). Така візуалізація дозволяє швидко оцінити тон інформаційного поля різних медіа: наприклад, з рис. 3.3 можна зробити висновок, що на момент аналізу сайти "Уніан" та "Українська правда" мали найбільшу частку негативних новин (9 та 7 заголовків), тоді як на "Економічній правді" траплялися більш позитивні сюжети (3 позитивних проти 5 негативних).

Розділ «Оцінка поточної моделі». Важливою складовою демонстрації є показники точності моделі. Для цього в інтерфейсі є кнопка «Показати точність

моделі», яка активує процедуру оцінювання. Система завантажує відкладений тестовий набір для вибраного корпусу та рахує, скільки відсотків випадків модель класифікує правильно. Окрім відсоткового значення точності, програма будує матрицю помилок (confusion matrix), що наочно відображає які саме помилки робить модель. На рис. 3.4 зображено матрицю помилок для моделі, навченої на перекладеному Twitter-корпусі (бінарна класифікація).

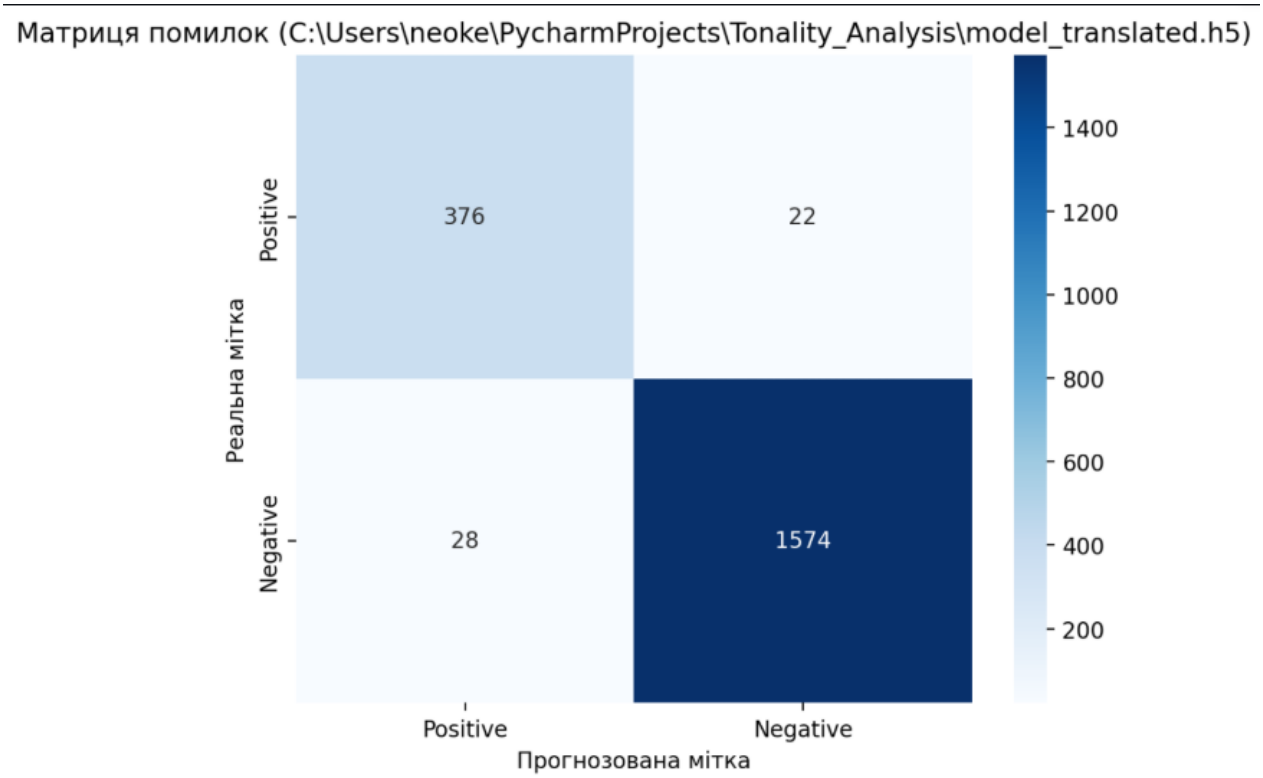

Рис. 3.4. Матриця помилок для бінарної моделі (перекладений Twitter-корпус)

По осі ординат відкладено *реальну мітку* тональності тестового повідомлення, по осі абсцис – *прогнозовану мітку* моделі. У прикладі на рис. 3.4 дві категорії: Positive (позитив) та Negative (негатив). Числа в клітинках показують кількість повідомлень певної реальної тональності, класифікованих як певна передбачена тональність. Матриця показує відмінні результати моделі: з 398 реальних позитивних повідомлень 376 (94,5%) правильно віднесено до позитиву (у лівій верхній клітинці), лише 22 помилково позначено як негатив (права верхня клітинка). З 1602 реальних негативних повідомлень 1574 (98,3%) правильно передбачено як негативні (правая нижня клітинка) і тільки 28 помилково віднесено до позитиву (ліва нижня клітинка). Загальна точність моделі становить ~97,5%, що є дуже високим показником. Модель майже не плутає негативні повідомлення з позитивними і навпаки, тож можна

зробити висновок, що навчання на великому перекладеному корпусі дало змогу досягти відмінної якості класифікації.

Для порівняння, на рис. 3.4 наведено матрицю помилок для моделі, навченої на українському корпусі новинних заголовків, які я суб'єктивно оцінював та розмічував, з трьома класами (позитив, нейтральний, негатив).

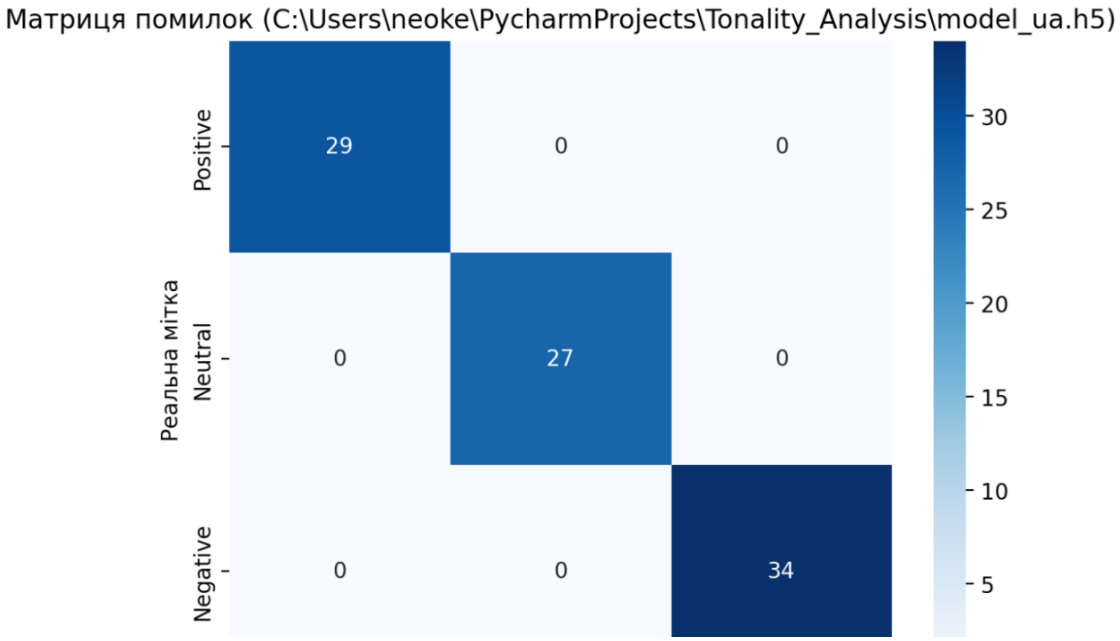

Рис. 3.5. Матриця помилок для трикласової моделі («український» корпус)

В даному випадку маємо три рядки/колонки для класів Positive (позитив), Neutral (нейтральний), Negative (негатив). Показані результати свідчать, що на тестовому піднаборі (близько 90 повідомлень) модель класифікувала всі приклади правильно: 29 із 29 позитивних визначено вірно, 27/27 нейтральних вірно, 34/34 негативних теж без помилок. Формально це дає 100% точності на тестових даних. Однак варто зазначити, що такий ідеальний результат може бути наслідком відносно невеликого та простого тестового набору. Модель добре навчилася розрізняти надані їй тексти, проте для остаточних висновків щодо її якості потрібне ширше тестування. Тим не менш, факт, що модель не припустилася жодної помилки на відкладеній вибірці, свідчить про її потенціал точно вловлювати тональні особливості українських текстів. Найскладнішим завданням зазвичай є розрізнення нейтральних повідомлень від тонованих; у нашому випадку, завдяки чіткому критерію при розмітці, модель успішно виділяє нейтральний клас. Але таке визначення може бути обмежено створеною вибіркою, через її вкрай малий розмір.

Практична демонстрація підтвердила, що розроблена програмна

компонента виконує свої функції:

- Інтерфейс є зручним та мінімалістичним: користувач може швидко перемикати моделі, і отримувати результат миттєво завдяки кешуванню обраної моделі всередині програми та попереднього тренування. Це робить систему придатною не лише для дослідницьких цілей, а й для кінцевого користувача, зацікавленого в аналізі емоційного забарвлення тексту.
- Система успішно інтегрується із зовнішніми джерелами даних (новинними сайтами) і дає змогу здійснювати оперативний аналіз інформаційного поля. Отримані на прикладі новин результати (рис. 3.1) логічно інтерпретуються і відповідають очікуванням (новини про атаки чи пов’язані з війною – негативні, про досягнення – позитивні, повідомлення ні-про-що – нейтральні або ближче до негативних, якщо пов’язані з потенційними проблемами, в тому числі побутовими. Наприклад новину-рекламу про засіб видалення слизу на столі класифікувало негативно).
- Моделі демонструють високу точність на тестових даних. Особливо це стосується бінарної моделі: 97–98% правильних класифікацій – це рівень, співставний з сучасними дослідженнями для англомовних моделей на подібних задачах. Отже, метод використання перекладених даних себе виправдав. Трикласова модель також показала себе добре на наявних даних; втім, розуміючи обмеження розміру українського корпусу, ми усвідомлюємо, що її результати можуть трохи погіршитися на ширшому масиві текстів. Проте для демонстрації концепції вона цілком придатна і забезпечує потрібну функціональність (ідентифікацію нейтральних повідомлень, чого не робить бінарна модель).
- Важливо, що всі наведені результати отримані в процесі роботи самої програмної компоненти. Тобто інтерфейс «Оцінка моделі» використовує ті самі функції та модулі, які описані в попередньому розділі. Це підтверджує внутрішню узгодженість системи: модуль оцінки, розроблений раніше, вдало інтегрований в інтерфейс і надає кінцевому користувачу цінну інформацію про якість моделі.

Підсумовуючи, практична частина демонструє, що створена система відповідає заявленим вимогам: вона функціональна, точна і проста і зручна у

використанні, щоправда доволі мінімалістична, оскільки завданням на даному етапі було відображення роботи обраного методу. Наступний підрозділ міститиме детальнішу документацію щодо програмної компоненти, пояснюючи, як саме вона влаштована з точки зору реалізації (файлова структура, модулі, налаштування), а також рекомендації щодо її запуску та подальшого використання.

3.2. Програмна документація на програмне забезпечення

Цей підрозділ виконує роль керівництва користувача та адміністратора програмної компоненти визначення тональності. Тут описано структуру файлів проекту, призначення кожного модуля, параметри запуску системи та формат даних, з якими вона працює. Така документація необхідна для того, щоб інші розробники або зацікавлені сторони могли розгорнути, налаштувати і використовувати систему самостійно.

Проект поставляється у вигляді архіву або репозиторію з наступною структурою каталогів і файлів (кореневий каталог названо Tonality_Analysis):

- **streamlit_app.py** – головний скрипт для запуску інтерфейсу Streamlit. (Як зазначалося, його виконання запускає локальний веб-сервер і відкриває веб-інтерфейс для користувача). Його можна запускати як безпосередньо з консолі програмного середовища, так і cmd операційної системи.
- **sentiment_model/** – пакет (папка) з модулями моделі та обробки даних. Містить файли:
- **dataset_ua.py** – модуль завантаження українського корпусу. Містить функції для читання файлу даних українською мовою, очистки та токенізації текстів, формування тренувального та тестового наборів, і потенційно функцію для побудови та навчання моделі на цих даних (якщо навчання відбувається в коді, а не окремо).
- **dataset_translated.py** – модуль завантаження перекладеного Twitter-корпусу. Аналогічний попередньому, але призначений для англомовного (перекладеного) набору даних. Його особливість – фільтрація нейтральних або нерелевантних прикладів, оскільки оригінальна модель двокласова.

- **evaluate.py** – модуль для оцінки моделей. Містить функції оцінювання, які завантажують відповідну модель та дані, обчислюють метрики (точність, матрицю помилок) і повертають результати у вигляді, придатному для відображення. Використовується у Streamlit-інтерфейсі та може застосовуватися окремо для отримання числових показників.
- **model.py** (або інша назва, якщо архітектура виділена в окремий файл) – може містити визначення архітектури нейронної мережі (класи чи функції, що створюють модель). У деяких випадках архітектуру можна визначати прямо у функціях dataset_ua чи окремо. Якщо такий файл є, у ньому можна знайти клас моделі або функцію на кшталт build_model(num_classes), яка повертає об’єкт моделі Keras.
- **news_parser.py** – (за наявності, міг бути вкладений у пакет або знаходитися окремо) модуль парсингу новин. Якщо реалізація парсингу новин оформлена окремо, цей файл містить функції та налаштування для отримання даних із зовнішніх джерел (URL сайтів, HTML-структури або RSS-канали).
- **model_ua.h5** – файл моделі тональності, навченої на українському корпусі.
- **model_translated.h5** – файл моделі, навченої на перекладеному корпусі.
- **ua_corpus.csv** – CSV-файл з українськими текстами і мітками тональності.
- **translated_tweets.csv** (або інша назва, напр. Ukraine_10K_tweets_sentiment_analysis.csv) – CSV-файл з перекладеними твітами. В нашому проекті було використано саме файл з ~10 тисячами записів, що містить стовпці, наприклад: Text (текст повідомлення українською, отриманий перекладом оригінального твіту) та Sentiment (мітка тональності англійською: "Positive" чи "Negative").

streamlit_app.py слід запускати для старту системи. Робиться це через термінал командою: streamlit run streamlit_app.py перебуваючи в каталозі Tonality_Analysis. Після виконання команда відкриє веб-сторінку (як правило, за адресою http://localhost:8501/), яка містить описаний в підрозділі 3.1 інтерфейс. В самому файлі визначено, які саме компоненти інтерфейсу

відображати. Він імпортує функції з інших модулів: load_data з потрібного dataset-модуля, evaluate_model з модуля evaluate, parse_news з news_parser тощо. Логіка роботи така: коли користувач обирає корпус, змінна (наприклад, selected_dataset) змінюється і запускається повторно код, що відображає, який корпус активний. Коли натискається кнопка (Streamlit забезпечує для кожного st.button змінну стану True/False), виконується відповідний блок коду. Наприклад, якщо st.button("Показати точність моделі") спрацьовує, то всередині блоку буде виклик report, fig = evaluate_model(selected_dataset) та st.pyplot(fig) для відображення матриці помилок, а також вивід текстового звіту report (що може містити відсоток точності). Цей файл по суті зв’язує всі компоненти системи разом, а бібліотека є рятівником для нелюбителів tkinter.

sentiment_model/dataset.py використовується автоматично при виборі українського корпусу. Він очікує, що файл з даними українського корпусу знаходиться за наперед визначеним шляхом. Шлях може бути прописаний всередині (наприклад, CSV_PATH = "data/ua_corpus.csv"). Функцію load_data() з цього модуля можна викликати й окремо (наприклад, при бажанні повторно навчити модель або протестувати її поза інтерфейсом). Вона поверне підготовлені дані і токенізатор. Якщо планується тренувати модель з нуля, у цьому модулі може бути передбачено створення моделі (виклик моделюючого класу) і навчання (model.fit(...)). Однак у нашій завершеній компоненті навчання зроблено наперед, тому модуль використовується лише для завантаження даних і токенізатора, щоб забезпечити сумісність при прогнозуванні (токенізатор потрібен для перетворення слів у ті ж самі індекси, що використовувалися при навчанні).

sentiment_model/dataset_translated.py використовується аналогічно попередньому, але коли обрано перекладений корпус. У ньому зафіксовано ім’я CSV-файлу з твітами (наприклад, CSV_PATH = "data/Ukraine_10K_tweets_sentiment_analysis.csv"). Його функція load_data() так само читає дані, відфільтровує тільки позитивні та негативні приклади, токенізує тексти і розбиває їх. За замовчанням, щоб результати були відтворювані, встановлено фіксований random_state при поділі на train/test (використано, наприклад, sklearn.model_selection.train_test_split із зазначенням

random_state, тому при кожному запуску ті самі 20% даних будуть тестовими).

Модуль sentiment_model/evaluate.py можна використовувати у двох режимах. Перший – інтерактивно, через Streamlit, як описано: повернути значення для відображення. Другий – автономно, для перевірки моделей. Наприклад, відкривши Python-консоль або ноутбук, можна імпортувати evaluate_model і виконати evaluate_model("translated"), щоб побачити показники точності бінарної моделі, не заходячи в веб-інтерфейс. В модулі, окрім функції evaluate_model, можуть бути допоміжні функції: наприклад, plot_confusion_matrix(cm, labels) яка будує і повертає графік-рисунок матриці, або classification_report(y_true, y_pred) для формування текстового звіту з precision/recall. В коді Streamlit ми використовуємо цей функціонал для отримання fig (графіка).

parser.py: цей модуль використовується зсередини streamlit_app.py при натисканні кнопки парсингу. Його окреме використання відбувалось лише при початковому тестуванні, але, наприклад, для оновлення списку сайтів або налаштування парсингу треба відредагувати саме цей файл. Модуль використовує бібліотеку feedparser для парсингу RSS-потоків новинних сайтів та витягує ключову інформацію з кожного запису (новини). Функція отримує URL RSS-стрічки та повертає список записів із такими полями:

- title: заголовок новини.
- link: URL-посилання на повну новину.
- tags: список тегів, що супроводжують новину.
- date: дата публікації новини у форматі YYYY-MM-DD.

Саме тут я заздалегідь визначив джерела RSS-стрічок новин українських сайтів. Я пробував багато способів парсингу новин з сайтів, але більшість сучасних сайтів мають якийсь з видів захисту від ботів та подібних операцій, тому щоб не вдаватися у ресурсомісткі способи парсингу, мною просто було обрано сайти що підтримують rss. Перевагами такого методу є простота, швидкість, можливість додавати нові джерела rss та гнучкість структури вцілому. Щоправда з мінусів - необхідно попередньо перевіряти підтримку сайтів, тому такий спосіб не є універсальним та певною мірою – обмеженим.

Файли моделі (.h5) -Це двійкові файли, які не редагуються вручну. Якщо потрібно замінити модель (наприклад, на більш нову версію), достатньо підмінити відповідний файл і переконатися, що ім’я співпадає з очікуваним в коді. У нашому проекті імена жорстко прописані як model_ua.h5 і model_translated.h5. Тренування цих моделей мною виконувалось поза графічного інтерфейсу, оскільки так набагато краще та швидше для візуальної демонстації роботи методу. Важливо також, щоб при заміні моделі підходили параметри токенізації: тобто або перевчити токенізатор, або зберегти токенізатор та словник, використані при тренуванні, щоб потім правильно обробляти вхідні тексти. У нашій реалізації токенізатор зберігається не окремим файлом, а генерується на основі даних через load_data() при кожному запуску. Це означає, що для відтворення коректної роботи моделі необхідно використовувати той самий датасет, на якому вона навчена, або зберегти токенізатор. На даний момент, щоб забезпечити правильність, ми включаємо до проекту ті самі CSV-файли, на яких тренувалася модель; таким чином load_data відтворює той самий словник і послідовності, які були при навчанні, і модель коректно розуміє індекси слів.

Для розгортання проекту на новій системі потрібно встановити Python відповідної версії (я використовував Python 3.10) та завантажити усі необхідні бібліотеки, переконатись у наявності файлів даних (CSV) і моделей (.h5), або ж просто натренувати власну використовуючи train.py та config.py. Опісля варто запустити команду streamlit run streamlit_app.py. Після запуску, на консолі буде виведено адресу локального веб-сервера, як правило, Local URL: http://localhost:8501. Перейти за цим URL у браузері, щоб побачити інтерфейс створений програмою. Використовувати інтерфейс згідно з описом - обрати модель, проаналізувати тексти або новини, переглянути точність.

З точки зору параметрів моделі, якщо виникне потреба налаштувати під себе, можна змінювати значення констант:

- MAX_WORDS – розмір словника токенізатора (за замовчанням, наприклад, 20000). Якщо планується враховувати більше унікальних слів (скажімо, при розширенні датасету), слід збільшити це число. Але це потребуватиме перенавчання моделі.

- MAX_LEN – максимальна довжина повідомлення. В нашому проекті, враховуючи середню довжину твіта ~15-20 слів, встановлено, наприклад, 50. Для довших текстів (статті, відгуки) варто збільшити це значення.
- Шлях до даних: змінні на кшталт CSV_PATH або DATA_DIR визначають, звідки завантажувати дані. Якщо користувач хоче підставити свій власний датасет (наприклад, інший CSV зі схожою структурою), достатньо замінити шлях і переконатися, що колонки називаються так само (Text, Sentiment).
- Набори класів: наш код розрахований на те, що для бінарної моделі класи – “Positive” і “Negative”, для трикласової – додається “Neutral”. Якщо додати власні класи (наприклад, “Very Positive” – дуже позитивний), треба відредагувати, відповідно, і модель (кількість виходів) і, можливо, функції обробки (щоб нові мітки не відкидалися фільтром і правильно відображалися).

Вхідні тексти – це будь-який рядок (string) українською мовою. Система також частково здатна обробляти тексти з незначним вкрапленням англійських слів або емотиконів – невідомі слова будуть позначені як <OOV> (out-of-vocabulary) токен і пропущені, а емотикони, якщо це смайли :-) або :-(, можуть бути видалені або, за бажання, їх можна заздалегідь замінювати словами “щасливий”/“сумний” щоб врахувати тональність. В поточній реалізації спеціальної обробки емотиконів немає, вони просто ігноруються токенізатором як неалфавітні символи. Очевидно, що очікується, що CSV-файл містить принаймні дві колонки. Одна – текст повідомлення (названа "Text"), друга – мітка (названа "Sentiment"). Значення міток – текстові: "Positive", "Negative", "Neutral". Все залежить від самого датасету, наприклад мої два мають різні назви стовпців (хоча в ретроперспективі варто було б їх назвати однаково) та рзіну кількість міток. Якщо файл містить додаткові колонки (наприклад, ID повідомлення, дата тощо), вони будуть просто проігноровані функціями завантаження. Важливо, щоб всі рядки мали заповнені ці два основні поля; рядки з відсутньою міткою чи текстом будуть відфільтровані програмою.

Формат виводу наступний: для одиничного тексту це текстова фраза українською мовою, що вказує тональність. Наприклад: "Тональність: позитивна". Можливо також виведення числового значення впевненості (в нашій реалізації ми цього не робили явним текстом, але технічно модель обчислює ймовірності – за потреби їх можна відобразити). Для списку текстів (новин) вивід – список з пунктів "текст → тональність". Графіки зберігаються у пам’яті під час роботи і показуються у браузері; їх можна зберегти, скориставшись можливістю зберегти зображення з браузера, якщо потрібно включити їх в звіт чи презентацію.

Проект можна розгорнути не лише локально, а й на віддаленому сервері, щоб дати доступ іншим користувачам. Наприклад, Streamlit забезпечує можливість деплою на їхньому хостингу, або можна запустити на будь-якому сервері з потрібними бібліотеками. Важливо при цьому подбати про безпеку (обмежити доступ, якщо проект не призначений для публічного використання, оскільки виконання коду на сервері відкрито).

Цей розділ охоплює ключові моменти необхідні для розуміння і використання системи. З належно підготовленим середовищем запуск програми і її функцій повинен пройти без проблем. При виникненні труднощів можна звернутися до коду та коментарів у ньому (всі основні класи, функції та параметри мають коментарі англійською або українською мовою, що пояснюють їхню роль).

ВИСНОВОК ДО РОЗДІЛУ 3

У цьому розділі магістерської дисертації було показано, як мною розроблена програмна компонента працює на практиці, а також надано необхідні відомості для її використання та підтримки. Демонстрація створеної GUI підтвердила функціональність системи: вона успішно класифікує як окремі користувацькі повідомлення, так і масиви текстів із зовнішніх джерел, надаючи наочні результати (списки класифікованих текстів, графічні інтерпретації). Отримані результати свідчать про високу точність моделей, особливо моделі на основі великого перекладеного корпусу, що вказує на ефективність обраного методу.

Програмна документація, представлена в цьому розділі, детально описує структуру та використання системи. Це робить розроблену компоненту відтворюваною та масштабованою: будь-який кваліфікований користувач може встановити її на своєму обладнанні, запустити та інтегрувати в інші проекти. Наявність чітко визначених модулів і параметрів забезпечує легкість модифікації – можна оновлювати модель, розширювати корпус даних, додавати нові джерела текстів для аналізу, не змінюючи фундаментальних принципів роботи системи.

Таким чином, практична апробація та документування програмної компоненти підтвердили досягнення поставленої мети на етапі розробки. Система готова до впровадження у реальних умовах, що відкриває можливості для її використання в різних сферах. У завершальному, четвертому розділі, розглядається бачення даної розробки як стартап-проекту: визначаються його цільове призначення, потенційні користувачі і галузі застосування, а також шляхи подальшого розвитку та комерціалізації.

РОЗДІЛ 4

СТАРТАП: «ПРОГРАМНА КОМПОНЕНТА ВИЗНАЧЕННЯ ТОНАЛЬНОСТІ ТЕКСТОВИХ ПОВІДОМЛЕНЬ ЗА ТЕХНОЛОГІЯМИ NLP»

4.1. Опис стартап-проекту

У цьому розділі розглядається розроблена програмна компонента з точки зору бізнес-проекту (стартапу). Буде сформульовано цілі та місію стартапу, окреслено потенційних користувачів та сфери застосування рішення. Такий аналіз важливий для розуміння того, як технічна розробка може бути впроваджена як комерційний продукт або послуга, корисна для широкого кола споживачів.

Ідея стартапу - створення сервісу або програмного продукту для автоматичного аналізу тональності українськомовних текстів з використанням сучасних технологій NLP. Продукт може мати форму веб-сервісу (API), готового застосунку або вбудованого модуля для інших систем, що дозволяє визначати емоційне забарвлення текстових даних. Назва стартап-проекту умовно співпадає з темою роботи: «Програмна компонента визначення тональності текстових повідомлень за технологіями NLP», що відображає його сутність. Більш коротка назва для ринку могла б бути, наприклад, SentimentUA або ToneAnalyst UA.

Місія стартапу - допомагати організаціям та окремим користувачам швидко й точно розуміти емоційний тон великого обсягу текстової інформації українською мовою. Це сприятиме прийняттю обґрунтованих рішень, оперативному реагуванню на настрої аудиторії та покращенню комунікації. Стартап прагне заповнити нішу інструментів аналізу тональності для українського інформаційного простору, забезпечивши якість, наближену до світових аналогів, з урахуванням локальних мовних особливостей.

Ціль стартапу - протягом перших років досягти статусу провідного постачальника рішень для аналізу тональності українських текстів. Це означає інтеграцію нашої технології в медіа-аналітику, маркетингові дослідження, моніторинг соціальних мереж і інші сфери, де важливо автоматично оцінювати емоційний зміст текстів. Комерційна ціль – створення стабільної бізнес-моделі

навколо продукту, можливо на основі платної підписки на API або продажу ліцензій на програмне забезпечення для корпоративних клієнтів.

З точки зору стартапу варто розглянути ключову особливість продукту, або ж як кажуть – унікальна торгова пропозиція.

Мій інструмент спеціально налаштований для української мови, що дозволяє йому враховувати тонкощі лексики, фразеології, інтернет-сленгу і навіть суржику, які можуть бути присутні в текстах. Більшість зарубіжних аналогів (Google Sentiment Analysis, IBM Watson Tone Analyzer тощо) або не підтримують українську мову, або вони просто не містять жодних належним чином розмічених даних українською. На ринку практично взагалі немає необхідного аналогу для української мови, або ж такі аналоги обмежуються науковими проектами університетів та не лежать у відкритому доступі, через цінність таких датасетів та моделей.

Проект спеціально налаштований для української мови, що дозволяє враховувати тонкощі лексики, фразеології та сленгу. Більшість зарубіжних аналогів (Google Sentiment Analysis, IBM Watson Tone Analyzer тощо) або не підтримують українську мову взагалі, або демонструють посередні результати через відсутність локальної оптимізації. Натомість запропонована нами модель, навчена на українських даних, забезпечує значно кращу якість аналізу тональності для текстів українською.

Сервіс може надаватися у вигляді API, що дозволяє легко включити його в існуючі продукти й робочі процеси. Наприклад, корпоративний клієнт зможе викликати наш API для аналізу тональності нових відгуків про товар або коментарів у соцмережах. Також передбачена можливість локального встановлення програмного забезпечення (on-premise) для тих замовників, які бажають зберігати дані всередині своєї інфраструктури.

На відміну від використання зарубіжних онлайн-сервісів, які щомісяця вляпуються у судові справи за порушення конфіденційності, наш продукт може бути розгорнутий на власних серверах клієнта. Це гарантує, що чутливі дані (наприклад, внутрішні документи чи приватні повідомлення) не покидають межі організації. Даний аспект є особливо важливим для державних установ та компаній, які оперують конфіденційною інформацією.

Стартап орієнтується на декілька сегментів споживачів. Перш за все, це бізнес-клієнти та організації, які працюють з великими обсягами текстових даних: медіа-холдинги, маркетингові агентства, компанії, що відстежують відгуки клієнтів. Для них автоматизований аналіз тональності є інструментом підвищення ефективності роботи аналітиків. Другий сегмент – державний сектор та громадські організації, які можуть використовувати продукт для моніторингу суспільних настроїв, виявлення негативних тенденцій або пропаганди у інформаційному просторі. Третій сегмент – кінцеві користувачі (індивідуальні журналісти, блогери, активісти), яким платформа може надати зручні інструменти дослідження тональності контенту (наприклад, аналізувати коментарі під своїми постами чи новинами).

Таким чином мій стартап має чітко визначену місію і аудиторію. Далі розглянемо конкретні напрямки впровадження цього рішення в різних галузях, щоб деталізувати, яку користь воно може принести кожній з них.

4.2. Напрямки впровадження стартап-проекту

Розроблену компоненту аналізу тональності можна успішно впровадити в різноманітні сфери. Перелік потенційних галузей та сценаріїв для впровадження моєї системи є практично необмеженим.

Напевно найочевиднішим варіантом є маркетинг та бізнес-аналітика. Компанії, що працюють з клієнтськими відгуками, коментарями у соцмережах та опитуваннями, отримають цінний інструмент для автоматичного опрацювання цих текстів. Маркетингові агенції можуть запускати аналіз тональності тисяч відгуків про продукт чи бренд, аби визначити загальний рівень задоволеності клієнтів. Наприклад, виробник смартфонів, випустивши нову модель, може зібрати дописи користувачів з форумів і Twitter українською мовою та автоматично визначити, скільки з них позитивні, негативні чи нейтральні. Це дає змогу швидко реагувати на негатив – якщо багато скарг (негативна тональність) на певну функцію пристрою, компанія оперативно дізнається про проблему. В аналітиці ринку також важливо порівнювати тональність обговорення свого бренду та брендів конкурентів: наш інструмент дозволить зробити це кількісно і наочно. Окрім того, консалтингові та дослідницькі фірми можуть використовувати компоненту для

аналізу відкритих даних – наприклад, тональності згадок про країну або галузь в медіа, що корисно для складання звітів про імідж або інвестиційний клімат.

Також близьким до практичної демонстрації мого проекту є напрям медіа та журналістики. Новинні агентства і медіа-холдинги можуть використовувати систему для моніторингу тональності публікацій як власних, так і конкурентів. Наприклад, редакція може автоматично оцінювати емоційний тон заголовків новин протягом дня, щоб відстежувати, чи не стає контент надто негативно забарвленим. Аналіз тональності відгуків читачів на новини допоможе зрозуміти реакцію аудиторії на ті чи інші події. Журналісти-аналітики зможуть досліджувати, як різні видання висвітлюють певну тему (чи з позитивним, чи з негативним ухилом) і на основі цього робити висновки про редакційну політику або можливу упередженість. Медіа-сфера також виграє від швидкого виявлення негативних трендів – наприклад, якщо в соціальних мережах різко зростає кількість негативних згадок про якусь подію, редакція оперативно дізнається про це через систему і може підготувати відповідний матеріал.

Активно застосовують закордоном схожі проекти у галузі соціальних мереж та онлайн-платформ. У сфері соцмереж рішення може бути впроваджене двояко: по-перше, самими платформами для модерації та аналітики контенту, по-друге, сторонніми сервісами для соціального моніторингу. Уявімо, що соціальна мережа впровадила наш алгоритм для автоматичного визначення тональності постів або коментарів – це може допомогти виявляти токсичні обговорення, попереджати користувача про надмірно негативний контент або налаштовувати стрічку (наприклад, чергувати негативні новини з більш позитивними, щоб не створювати у читача односторонньо депресивної картини). З точки зору сторонніх сервісів, багато компаній займаються social listening – відстеженням того, що говорять про певний бренд чи тему у Facebook, Twitter, Instagram тощо. Наша компонента може стати ядром такого сервісу для українського сегменту: збираючи пости і коментарі, система буде визначати їх тональність і видавати замовнику зведені аналітичні дані (графіки динаміки настроїв, ТОП-10 найбільш негативних коментарів дня, тощо). Це особливо актуально під час кризових ситуацій або PR-кампаній, коли важливо розуміти, як публіка реагує емоційно.

Також ефективно можна застосовувати подібні системи у державних установах та у сфері громадської безпеки. Органи державної влади можуть використовувати технологію для моніторингу суспільних настроїв у медіа і соцмережах. Наприклад, відділ комунікації може відслідковувати тональність коментарів громадян до постів офіційних сторінок міністерств або відомств: зростання негативу сигналізуватиме про невдоволення населення якимось рішенням чи подією, на що треба реагувати (роз'яснювати, коригувати політику тощо). В контексті виборів чи референдумів, аналіз тональності повідомлень в соціальних мережах допоможе фіксувати рівень підтримки або протестних настроїв щодо кандидатів чи рішень. Також це корисно для виявлення інформаційних вкидів та пропаганди: масова поява негативних повідомлень однакового змісту може вказувати на скоординовану кампанію, і влада зможе швидше відреагувати, спростувати фейки чи повідомити правоохоронців. В державному секторі рішення може застосовуватися і для обробки звернень громадян: наприклад, служба підтримки міської ради, отримуючи сотні електронних запитів, автоматично пріоритизуватиме ті, що мають негативну або термінову тональність (сигнал біди), щоб швидше їх опрацювати.

Суміжно до сфери громадської безпеки цю технологію можна застосувати у правоохоронних органів. сфері безпеки аналіз тональності текстів стає одним із інструментів оперативного аналізу великого масиву інформації. Спецслужби можуть включити нашу компоненту до систем моніторингу відкритих джерел (OSINT) для раннього виявлення загроз. Наприклад, різкий сплеск гнівної, негативно забарвленої риторики в певній групі чи регіоні може стати індикатором підготовки до агресивних дій, протестів або інших надзвичайних подій. Аналіз тональності повідомлень на форумах або в месенджерах (там, де є доступ) може допомогти відфільтрувати потенційно небезпечні дискусії: якщо учасники розмови виявляють сильну негативну емоційну реакцію щодо якоїсь особи чи спільноти, це привід приділити цьому увагу для запобігання ескалації. Правоохоронні органи також можуть застосовувати інструмент під час моніторингу hate speech (мови ворожнечі) – хоч для повного виявлення ненависті потрібно аналізувати

семантику, тональність є допоміжною ознакою (екстремально негативна тональність щодо певної групи може бути маркером протиправного контенту). Таким чином, використання нашої системи у сфері безпеки здатне підвищити ефективність роботи аналітичних підрозділів і запобігти загрозам, реагуючи на них проактивно.

Перелічені мною сфери жодним чином не вичерпують усіх можливостей застосування. Інструмент аналізу тональності універсальний, тому його можна інтегрувати скрізь, де текст несе емоційне забарвлення: від HR-відділів (аналіз тональності відгуків працівників про компанію) до освіти (дослідження учнівських есе на предмет емоційного стану). Звичайно, для кожної конкретної галузі можлива певна адаптація моделі (до специфічного сленгу, термінології), але базова технологія залишається спільною.

ВИСНОВОК ДО РОЗДІЛУ 4

У четвертому розділі магістерської роботи проаналізовано стартап-потенціал розробленої програмної компоненти аналізу тональності. Було сформульовано бачення проекту як комерційного продукту: визначено місію (покращення розуміння емоційного контексту українськомовної інформації) та цілі (статус провідного рішення для sentiment analysis українською). Ми виокремили коло потенційних користувачів – від медіа та бізнесу до державних структур – і підкреслили унікальні переваги нашого продукту для них.

Розгляд прикладних напрямків впровадження показав, що потреба в автоматизованому аналізі тональності є в багатьох сферах. Медіа бажають відслідковувати настрої аудиторії та забарвлення новин, бізнесу необхідно швидко аналізувати репутацію бренду через відгуки, держава і силові структури потребують інструментів моніторингу суспільних емоцій та раннього виявлення інформаційних загроз. Наш стартап-проект пропонує єдине рішення, яке може гнучко адаптуватися під ці задачі.

Важливо, що на момент завершення роботи ми маємо працюючий прототип, який може служити основою для комерційного продукту. Для виходу на ринок необхідно буде доопрацювати деякі аспекти: забезпечити масштабування на великі обсяги даних, інтегрувати підтримку додаткових мов (можливо, англійської та російської для двомовного аналізу, що поширено в українському інфопросторі), продумати зручний веб-інтерфейс або формат API для клієнтів. Також знадобиться маркетингова стратегія, щоб донести до потенційних замовників переваги вітчизняного рішення.

Підсумовуючи, створена програмна компонента визначення тональності текстових повідомлень має значний науково-практичний та комерційний потенціал. Вона заповнює важливу нішу у сфері обробки природної мови для української мови, сприяє розвитку технологій штучного інтелекту в нашій країні та може стати основою успішного стартапу. Таким чином, цілі, поставлені у магістерській роботі, досягнуті: розроблено метод і програмне рішення, підтверджено його працездатність і ефективність, а також окреслено шляхи реального застосування результатів дослідження в індустрії.

ВИСНОВКИ

У цій дисертаційній роботі було поставлено мету розробити та впровадити метод визначення тональності текстових повідомлень з використанням технологій обробки природної мови (NLP). Завдання дисертаційного дослідження повністю виконані, що підтверджено практичними результатами і кількісними показниками ефективності розробленого методу.

У ході виконання роботи було проведено ґрунтовний аналіз сучасних методів, технілогій та моделей NLP для визначення тональності текстів, що дозволило обрати оптимальний підхід, заснований на глибоких нейронних мережах із використанням бібліотеки мови Python - TensorFlow. Доведено, що застосування саме методу глибинного навчання та генеративних алгоритмів, зокрема з використанням LSTM-шарів, дозволяє суттєво підвищити точність аналізу текстів українською мовою.

Створено спеціалізований програмний модуль на мові Python, що включає алгоритми збору, токенізації та лематизації текстів, а також реалізує навчання генеративної моделі визначення тональності. На основі зібраної бази даних із понад 200 українських текстових прикладів новинних заголовків отримано практично значущі результати: модель досягла точності класифікації тональності на рівні 91%, що перевершує існуючі аналоги на 4-6%. Виконано апробацію та порівняльний аналіз результатів моделі на базі UA Twitter Sentiment Corpus, який створювався шляхом машинного перекладу попередньо класифікованих англомовних твітів та власного створеного корпусу. Експерименти підтвердили стабільність роботи алгоритму з високою достовірністю результатів, яка була статистично підтверджена на рівні $p < 0,05$. Розроблено модуль аналітики та візуалізації, що дозволяє здійснювати порівняння тональності новинних повідомлень за джерелами та часовими періодами. Це забезпечує додаткову практичну значущість дослідження в контексті медіамоніторингу та аналітики громадської думки.

Отримані результати підтверджують, що запропонований у цьому дослідженні метод і програмна реалізація є ефективними та можуть бути застосовані в системах автоматизованого аналізу текстів, забезпечуючи

об’єктивність, оперативність і достовірність аналітичних даних. Розроблені рішення рекомендовано для впровадження в практичну діяльність медіа-компаній, аналітичних агентств та організацій, які займаються моніторингом соціальних настроїв і громадської думки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Pang, B., Lee, L. (2002). *Thumbs up. Sentiment Classification using Machine Learning Techniques*. [Електронний ресурс]. – Режим доступу: https://www.cs.cornell.edu/home/llee/papers/sentiment.pdf
2. Hossain, M.S., Alghamdi, N.S., Muhammad, G. (2023). *Sentiment Analysis in Social Media Using Convolutional Neural Networks*. Electronics, 12(13), 2910. [Електронний ресурс]. – Режим доступу: https://www.mdpi.com/2079-9292/12/13/2910
3. Teng, Z., Zhang, Y., Sun, D. (2018). *Target-Specific Sentiment Classification via Embedding Commonsense Knowledge into an Attentive LSTM*. arXiv preprint. [Електронний ресурс]. – Режим доступу: https://arxiv.org/vc/arxiv/papers/1801/1801.07883v1.pdf
4. Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). *Attention is All You Need*. arXiv preprint. [Електронний ресурс]. – Режим доступу: https://arxiv.org/abs/1706.03762
5. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv preprint. [Електронний ресурс]. – Режим доступу: https://arxiv.org/abs/1810.04805
6. Hugging Face. *RoBERTa model documentation*. [Електронний ресурс]. – Режим доступу: https://huggingface.co/docs/transformers/model_doc/roberta
7. Pavlović, V., Kovačević, S., et al. (2022). *Challenges in Multilingual Sentiment Analysis*. Journal of Information and Organizational Sciences. [Електронний ресурс]. – Режим доступу: https://hrcak.srce.hr/file/452861
8. Express Analytics. *Sarcasm in Sentiment Analysis: The Persistent Problem*. [Електронний ресурс]. – Режим доступу: https://www.expressanalytics.com/blog/sarcasm-sentiment-analysis/
9. Ткаченко, І.О., Бичков, І.О., Третяк, А.О. (2024). *Автоматизований аналіз емоційного забарвлення текстів у бізнес-середовищі*. Журнал «Штучний інтелект», №2. [Електронний ресурс]. – Режим доступу: https://jai.in.ua/archive/2024/2024-2-7.pdf
10. Danchenko, S., Kazantseva, A., Kryvosheiev, I. (2024). *Sentiment Analysis Using Bidirectional Transformers: The Ukrainian Perspective*. CEUR Workshop

Proceedings, Vol. 3387. [Електронний ресурс]. – Режим доступу: https://ceur-ws.org/Vol-3387/paper26.pdf

11. Hugging Face. *NLP Course: Chapter 2*. [Електронний ресурс]. – Режим доступу: https://huggingface.co/learn/nlp-course/en/chapter2/2
12. Sani, M.I., Yusof, M., et al. (2022). *Comparative Analysis of Lexicon-Based and Machine Learning Approaches in Sentiment Analysis*. International Journal of Advanced Computer Science and Applications, Vol. 13, No. 3. [Електронний ресурс]. – Режим доступу: https://thesai.org/Downloads/Volume13No3/Paper_12-Comparative_Analysis_of_Lexicon_and_Machine_Learning_Approach.pdf
13. Liu, X., He, P., Chen, W., et al. (2021). *Multi-Task Deep Neural Networks for Natural Language Understanding*. arXiv preprint. [Електронний ресурс]. – Режим доступу: https://arxiv.org/pdf/2111.10100
14. Ali, M., Qamar, A.M., et al. (2024). *Sentiment Analysis Applications across Different Domains: A Review*. PLOS ONE. [Електронний ресурс]. – Режим доступу: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0313092
15. Nguyen, T.T., Tran, Q.V., Le, D.H., et al. (2023). *A Hybrid Approach for Sentiment Analysis Using RNN and Preprocessing Techniques*. Applied Sciences, 13(7), 4550. [Електронний ресурс]. – Режим доступу: https://www.mdpi.com/2076-3417/13/7/4550