Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА

на тему *Метод класифікації настроїв у соціальних мережах з використанням обробки природної мови*

Рівень вищої освіти *другий (магістерський)*

Галузь знань *12 – Інформаційні технології*
Шифр і найменування

Спеціальність *122 – Комп'ютерні науки*
Код і найменування

Освітня програма *Комп'ютерні науки*
Назва

Виконав: *студент 2 курсу, група КНм-24-1* *Богдан ВОЛКОЛУП*
Курс, група виконавця — Підпис — Ім'я, ПРІЗВИЩЕ

Керівник: *к.т.н., доцент кафедри КН* *Олександр ПАСІЧНИК*
Науковий ступінь, посада — Підпис — Ім'я, ПРІЗВИЩЕ

Нормоконтроль: *к.т.н., доцент кафедри КН* *Руслан БАГРІЙ*
Науковий ступінь, посада — Підпис — Ім'я, ПРІЗВИЩЕ

До захисту допускаю:

Зав. кафедри КН, д.т.н., професор *Олександр БАРМАК*
Підпис — Ім'я, ПРІЗВИЩЕ

12 грудня 2025 р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Факультет *інформаційних технологій*
Кафедра *комп'ютерних наук*
Освітній ступінь *магістр*
Галузь знань *12 – Інформаційні технології*
Спеціальність *122 – Комп'ютерні науки*

ЗАТВЕРДЖУЮ
Завідувач кафедри *комп'ютерних наук*
_____
(підпис)
д.т.н., професор Олександр БАРМАК
«*28*» *08* *2025* року

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

1. Тема кваліфікаційної роботи: *«Метод класифікації настроїв у соціальних мережах з використанням обробки природної мови»*
2. Завдання видано студенту *Богдану ВОЛКОЛУПУ*
(Ім'я, ПРІЗВИЩЕ)
3. Керівник роботи *доцент кафедри КН Олександр ПАСІЧНИК*
(Ім'я, ПРІЗВИЩЕ)
4. Затверджені наказом університету від «*25*» *08* *2025* р. № *65*
5. Дата видачі завдання студенту: «*28*» *08* *2025* р.
6. Зміст пояснювальної записки (перелік задач) та вихідні дані:

Мета роботи полягає у підвищенні точності класифікації емоційного забарвлення текстових повідомлень у соціальних мережах шляхом розробки методу з використанням рекурентних нейронних мереж LSTM та спеціалізованої обробки специфічних елементів онлайн-комунікації. Задачі: проаналізувати існуючі підходи до класифікації настроїв на основі методів машинного навчання; розробити метод класифікації емоційного забарвлення текстів з використанням архітектури LSTM; провести експериментальне дослідження ефективності запропонованого методу шляхом порівняння з базовими алгоритмами класифікації.

Ключові слова: аналіз настроїв, соціальні мережі, LSTM, рекурентні нейронні мережі, обробка природної мови, емодзі, класифікація тексту

7. Календарний план виконання кваліфікаційної роботи:

№	Назва етапів (розділів) кваліфікаційної роботи магістра	Термін виконання	Примітка
1	*Вибір напряму дослідження та узгодження теми кваліфікаційної роботи з керівником, складання календарного графіка виконання роботи*	*вересень 2025*	*Виконано*
2	*Ознайомлення з предметною областю, аналіз існуючих методів і моделей, формулювання мети та завдань дослідження, визначення об'єкта й предмета дослідження*	*вересень 2025*	*Виконано*
3	*Розробка методу чи моделі для вирішення обраного завдання, опис архітектури рішення*	*жовтень 2025*	*Виконано*
4	*Програмна реалізація методу чи моделі*	*жовтень 2025*	*Виконано*
5	*Дослідження ефективності та експериментальна перевірка результатів, порівняння з відомими підходами*	*листопад 2025*	*Виконано*
6	*Написання пояснювальної записки, оформлення відповідно до вимог, врахування зауважень керівника*	*листопад 2025*	*Виконано*
7	*Підготовка презентаційних матеріалів та попередній захист*	*листопад 2025*	*Виконано*
8	*Перевірка пояснювальної записки на відповідність вимогам оформлення (нормоконтроль) та перевірка на академічну доброчесність. Отримання відгуку керівника та рецензії.*	*грудень 2025*	*Виконано*
9	*Публічний захист кваліфікаційної роботи*	*грудень 2025*	*Виконано*

Виконавець: *студент групи КНм-24-1* _____ *Богдан ВОЛКОЛУП*
Група виконавця — Підпис — Ім'я, ПРІЗВИЩЕ

Керівник: *к.т.н., доцент каф. КН* _____ *Олександр ПАСІЧНИК*
Науковий ступінь, посада — Підпис — Ім'я, ПРІЗВИЩЕ

Реферат

Кваліфікаційна робота магістра присвячена розробці методу класифікації настроїв у соціальних мережах з використанням машинного навчання.

Актуальність теми. Актуальність роботи обумовлена зростаючою потребою у автоматизованому аналізі великих обсягів текстової інформації з соціальних платформ із застосуванням сучасних технологій штучного інтелекту. Традиційні методи аналізу настроїв, які базуються на лексичних словниках та простих статистичних підходах, є недостатньо ефективними, потребують значних ресурсів для підтримки, схильні до помилок при роботі з неформальною мовою і не враховують контекстуальні особливості онлайн-комунікації.

Сучасні досягнення дозволяють суттєво підвищити якість і точність визначення емоційного забарвлення текстів, автоматизуючи процеси та забезпечуючи здатність моделей адаптуватися до специфіки різних соціальних платформ. Це сприяє кращому розумінню суспільних настроїв, підвищенню ефективності моніторингу репутації брендів, поліпшенню якості обслуговування клієнтів та оптимізації маркетингових стратегій на основі аналізу відгуків користувачів.

Застосування векторних представлень слів та спеціалізованих підходів до обробки емодзі і сленгу допомагає подолати обмеження, пов'язані з неформальністю та специфічністю мови соціальних мереж, що є типовою проблемою в цій предметній області. Це підкреслює актуальність роботи як для практичних застосувань у бізнесі та маркетингу, так і для наукових досліджень у галузі застосування нейромережевих підходів до аналізу природної мови.

Мета роботи полягає у підвищенні точності класифікації емоційного забарвлення текстових повідомлень у соціальних мережах шляхом розробки методу з використанням рекурентних нейронних мереж LSTM та спеціалізованої обробки специфічних елементів онлайн-комунікації.

Задачі дослідження:

– провести аналіз існуючих методів та підходів до класифікації настроїв у текстах з використанням методів машинного та глибокого навчання;

– розробити метод класифікації емоційного забарвлення з використанням рекурентних нейронних мереж на основі архітектури LSTM з модифікаціями для обробки емодзі та сленгу;

– здійснити програмну реалізацію методу класифікації емоційного забарвлення текстових повідомлень у соціальних мережах;

– провести експериментальне дослідження ефективності спроектованого методу шляхом порівняння з базовими алгоритмами класифікації та оцінки його точності на тестових даних.

Об'єкт дослідження – процес визначення емоційного забарвлення текстових повідомлень у соціальних мережах.

Предмет дослідження – моделі, методи та засоби класифікації настроїв на основі рекурентних нейронних мереж з обробкою емодзі та сленгових виразів.

Методи дослідження. Застосовано рекурентні нейронні мережі, методи векторизації тексту, токенізації, нормалізації даних, дропаут-регуляризації, експериментальне тестування на реальних наборах даних з соціальних платформ.

Наукова новизна одержаних результатів. Удосконалено метод класифікації настроїв у текстах соціальних мереж, який відрізняється від існуючих використанням рекурентних нейронних мереж типу LSTM з додатковим етапом попередньої обробки, що включає побудову сентимент-орієнтованих словників для відображення емодзі на текстові дескриптори емоційних станів та лексичну нормалізацію сленгових конструкцій, що дозволило підвищити точність класифікації на 3.5% порівняно з базовою моделлю.

Апробація результатів кваліфікаційної роботи магістра та публікації. Волколуп Б.А., Пасічник О.А., Скрипник Т.К. Метод класифікації настроїв у текстах соціальних мереж на основі рекурентних нейронних мереж. Збірник наукових праць за матеріалами XVII Всеукраїнської науково-практичної конференції «Актуальні проблеми комп’ютерних наук АПКН-2025». - Хмельницький, 2025. - С.72 – 74.

Структура та обсяг роботи. Робота містить вступ, чотири розділи, висновки, список використаних джерел та додатків. Обсяг основного тексту становить 86 сторінки, включаючи 20 рисунків, 2 таблиці та переліку посилань з 45 джерел.

Ключові слова: аналіз настроїв, соціальні мережі, LSTM, рекурентні нейронні мережі, обробка природної мови, Word2Vec, емодзі, сленг, класифікація тексту, глибоке навчання.

Зміст

Перелік скорочень

Скорочення, термін, позначення	**Пояснення**
LSTM	Long Short-Term Memory (Довга короткочасна пам'ять)
RNN	Recurrent Neural Network (Рекурентна нейронна мережа)
NLP	Natural Language Processing (Обробка природної мови)
МН	Машинне навчання
ШІ	Штучний інтелект
CNN	Convolutional Neural Network (Згорткова нейронна мережа)
TF-IDF	Term Frequency-Inverse Document Frequency
SVM	Support Vector Machine (Метод опорних векторів)
ReLU	Rectified Linear Unit (Випрямлена лінійна одиниця)
Word2Vec	Word to Vector (Векторне представлення слів)
GloVe	Global Vectors for Word Representation
API	Application Programming Interface

Вступ

Кваліфікаційна робота магістра присвячена розробці методу класифікації настроїв у соціальних мережах з використанням машинного навчання.

Актуальність теми. Актуальність роботи обумовлена зростаючою потребою у автоматизованому аналізі великих обсягів текстової інформації з соціальних платформ із застосуванням сучасних технологій штучного інтелекту. Традиційні методи аналізу настроїв, які базуються на лексичних словниках та простих статистичних підходах, є недостатньо ефективними, потребують значних ресурсів для підтримки, схильні до помилок при роботі з неформальною мовою і не враховують контекстуальні особливості онлайн-комунікації. Сучасні досягнення дозволяють суттєво підвищити якість і точність визначення емоційного забарвлення текстів, автоматизуючи процеси та забезпечуючи здатність моделей адаптуватися до специфіки різних соціальних платформ. Це сприяє кращому розумінню суспільних настроїв, підвищенню ефективності моніторингу репутації брендів, поліпшенню якості обслуговування клієнтів та оптимізації маркетингових стратегій на основі аналізу відгуків користувачів. Застосування векторних представлень слів та спеціалізованих підходів до обробки емодзі і сленгу допомагає подолати обмеження, пов'язані з неформальністю та специфічністю мови соціальних мереж, що є типовою проблемою в цій предметній області. Це підкреслює актуальність роботи як для практичних застосувань у бізнесі та маркетингу, так і для наукових досліджень у галузі застосування нейромережевих підходів до аналізу природної мови.

Мета роботи полягає у підвищенні точності класифікації емоційного забарвлення текстових повідомлень у соціальних мережах шляхом розробки методу з використанням рекурентних нейронних мереж LSTM та спеціалізованої обробки специфічних елементів онлайн-комунікації.

Задачі дослідження:

– провести аналіз існуючих методів та підходів до класифікації настроїв у текстах з використанням методів машинного та глибокого навчання;

– розробити метод класифікації емоційного забарвлення з використанням рекурентних нейронних мереж на основі архітектури LSTM з модифікаціями для обробки емодзі та сленгу;

– здійснити програмну реалізацію методу класифікації емоційного забарвлення текстових повідомлень у соціальних мережах;

– провести експериментальне дослідження ефективності спроектованого методу шляхом порівняння з базовими алгоритмами класифікації та оцінки його точності на тестових даних.

Об'єкт дослідження – процес визначення емоційного забарвлення текстових повідомлень у соціальних мережах.

Предмет дослідження – моделі, методи та засоби класифікації настроїв на основі рекурентних нейронних мереж з обробкою емодзі та сленгових виразів.

Методи дослідження. Застосовано рекурентні нейронні мережі, методи векторизації тексту, токенізації, нормалізації даних, дропаут-регуляризації, експериментальне тестування на реальних наборах даних з соціальних платформ.

Наукова новизна одержаних результатів. Удосконалено метод класифікації настроїв у текстах соціальних мереж, який відрізняється від існуючих використанням рекурентних нейронних мереж типу LSTM з додатковим етапом попередньої обробки, що включає побудову сентимент-орієнтованих словників для відображення емодзі на текстові дескриптори емоційних станів та лексичну нормалізацію сленгових конструкцій, що дозволило підвищити точність класифікації на 3.5% порівняно з базовою моделлю.

Апробація результатів кваліфікаційної роботи магістра та публікації. Волколуп Б.А., Пасічник О.А., Скрипник Т.К. Метод класифікації настроїв у текстах соціальних мереж на основі рекурентних нейронних мере. Збірник наукових праць за матеріалами XVII Всеукраїнської науково-практичної конференції «Актуальні проблеми комп’ютерних наук АПКН-2025». - Хмельницький, 2025. - С.72 – 74.

Структура та обсяг роботи. Робота містить вступ, чотири розділи, висновки, список використаних джерел та додатків. Обсяг основного тексту становить 86 сторінки, включаючи 20 рисунків, 2 таблиці та переліку посилань з 45 джерел.

Ключові слова: аналіз настроїв, соціальні мережі, LSTM, рекурентні нейронні мережі, обробка природної мови, Word2Vec, емодзі, сленг, класифікація тексту, глибоке навчання.

Розділ 1 Дослідження проблеми класифікації настроїв у соціальних мережах

1.1 Характеристика задачі класифікації настроїв у соціальних мережах

Класифікація настроїв у текстах соціальних мереж належить до предметної області обробки природної мови та аналізу великих обсягів даних. Основна мета полягає в автоматичному визначенні емоційного забарвлення повідомлень, що публікуються користувачами на платформах Твіттер, Фейсбук, Інстаграм та інших соціальних мережах. Ця задача має велике практичне значення для бізнесу, моніторингу громадської думки та виявлення кризових ситуацій.

Соціальні мережі генерують величезні обсяги текстових даних, які мають специфічні характеристики. Повідомлення в Твіттері обмежені певною кількістю символів, що призводить до використання скорочень, емодзі та хештегів. Користувачі часто застосовують неформальну мову, сленг та абревіатури, що ускладнює стандартні підходи до обробки тексту. Ці особливості вимагають спеціальних методів попередньої обробки та векторизації текстових даних [1, 2].

Процес класифікації настроїв включає декілька етапів. Спочатку виконується попередня обробка тексту, де застосовуються алгоритми токенізації для розбиття тексту на окремі слова або токени. Бібліотека NLTK часто використовується для цієї мети з налаштуваннями для англійської мови, хоча для інших мов можна застосовувати бібліотеку SpaCy. Після токенізації виконується нормалізація тексту, що включає видалення стоп-слів за спеціальними словниками та лематизацію для приведення слів до їх базової форми. Ці кроки дозволяють зменшити розмірність даних та зберегти важливу семантичну інформацію [3, 4].

Векторизація тексту є критично важливим етапом, оскільки алгоритми машинного навчання працюють з числовими представленнями. Метод TF-IDF часто застосовується для перетворення тексту в числові вектори, де враховується важливість слів у документі відносно всього корпусу текстів. Альтернативним підходом є використання векторних представлень слів, таких як Word2Vec або GloVe,

які захоплюють семантичні зв'язки між словами. Ці методи створюють щільні векторні представлення, де схожі за значенням слова розташовані близько в векторному просторі [3, 5].

Задача класифікації може бути сформульована як бінарна, де визначається тільки позитивний або негативний настрій, або як багатокласова, де додатково виділяється нейтральний клас. У деяких дослідженнях розглядається більш детальна класифікація емоцій, включаючи радість, гнів, сум та страх. Вибір типу класифікації залежить від конкретної задачі та доступних анотованих даних [6, 7].

Важливою проблемою є незбалансованість класів у датасетах соціальних мереж. Часто нейтральні повідомлення переважають над емоційно забарвленими, що може призвести до зміщення моделі в бік більш представленого класу. Для вирішення цієї проблеми застосовуються техніки балансування, такі як надвибірка, недовибірка або синтетична генерація прикладів методом синтетичної надвибірки меншості. Ці підходи дозволяють покращити якість класифікації для менш представлених класів [8–10].

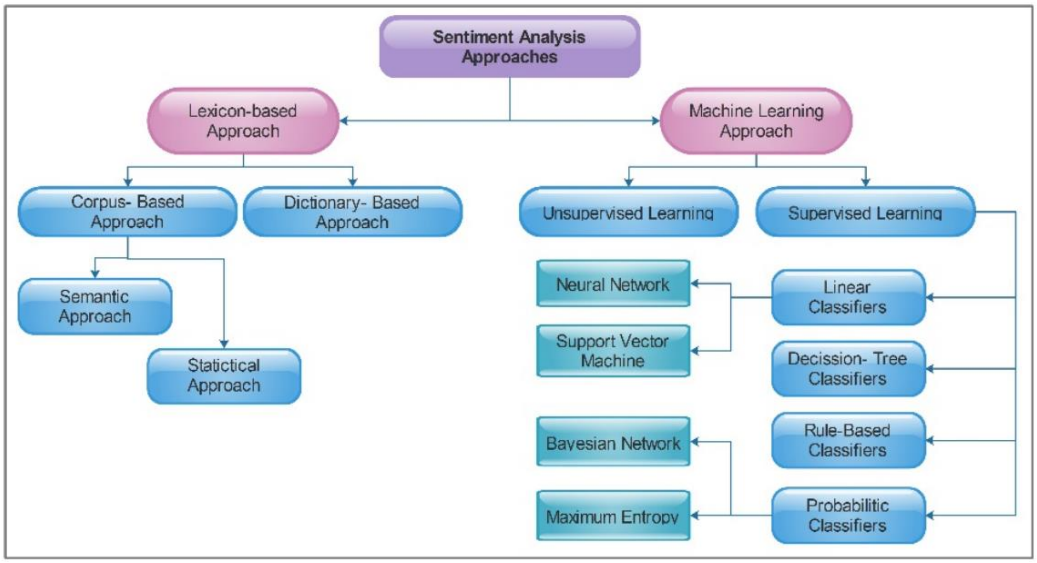


Рисунок 1.1 – Широко використовувані підходи до аналізу настроїв [8]

Для оцінки ефективності моделей класифікації застосовуються різні метрики. Точність показує загальну частку правильно класифікованих прикладів, але може бути оманливою на незбалансованих даних. Метрика прецизійності вимірює частку

правильно визначених добрих прикладів серед всіх зразків, класифікованих як позитивні. Повнота показує, яку частину позитивних прикладів вдалося знайти. F1-оцінка є гармонійним середнім між прецизійністю та повнотою, що робить її корисною для незбалансованих датасетів. У дослідженнях на датасетах Kaggle моделі часто досягають F1-оцінки на рівні 0.85-0.90 на контрольованих даних, але ефективність знижується на реальних шумних текстах [11–13].

Суттєвим викликом є обробка сарказму та іронії в текстах соціальних мереж. Ці лінгвістичні явища призводять до того, що буквальне значення тексту не відповідає справжньому настрою автора. Стандартні методи класифікації часто помиляються на таких прикладах, що знижує загальну точність системи. Для покращення обробки сарказму дослідники використовують контекстну інформацію, аналіз метаданих та спеціалізовані архітектури нейронних мереж з механізмами уваги [14, 15].

Мультимодальність є ще одним аспектом задачі класифікації настроїв у соціальних мережах. Користувачі доволі часто пишуть повідомлення, що містять не тільки текст, але й зображення, відео або інші медіа-елементи. Комплексний аналіз всіх цих модальностей може значно покращити якість визначення настрою. Проте інтеграція різних типів даних вимагає складніших архітектур моделей та більших обчислювальних ресурсів [1, 7, 16].

Обробка тексту в реальному часі ставить додаткові вимоги до швидкості роботи алгоритмів. Для моніторингу соціальних мереж часто необхідно обробляти потоки повідомлень з мінімальною затримкою. Це вимагає оптимізації моделей для швидкого виведення, що може включати квантизацію ваг нейронної мережі, використання ефективних архітектур або застосування апаратного прискорення [7, 17].

Етичні аспекти також відіграють важливу роль в задачі класифікації настроїв. Моделі можуть мати систематичні упередження, якщо навчальні дані не є достатньо репрезентативними. Упередження можуть проявлятися у формі гіршої якості для

певних демографічних груп або мовних варіантів. Дослідники приділяють увагу розробці справедливих моделей та методів виявлення і усунення упереджень [18, 19].

Для багатомовної класифікації настроїв виникають специфічні виклики. Більшість наявних датасетів та моделей створені для англійської мови, тому їх застосування до інших мов вимагає адаптації. Трансферне навчання дозволяє використовувати знання, отримані на великих англомовних корпусах, для покращення якості на мовах з меншою кількістю анотованих даних. Багатомовні моделі можуть працювати з декількома мовами одночасно, що робить їх корисними для глобальних платформ соціальних мереж [19, 20].

Практичне застосування методів класифікації настроїв включає моніторинг репутації брендів, аналіз відгуків клієнтів, виявлення кризових ситуацій та прогнозування трендів. Компанії використовують ці технології для швидкої зміни реакції щодо поганих відгуки та підтримки позитивного іміджу в соціальних мережах. Різні організації можуть застосовувати аналіз настроїв для оцінки громадської думки щодо політичних рішень [21, 22].

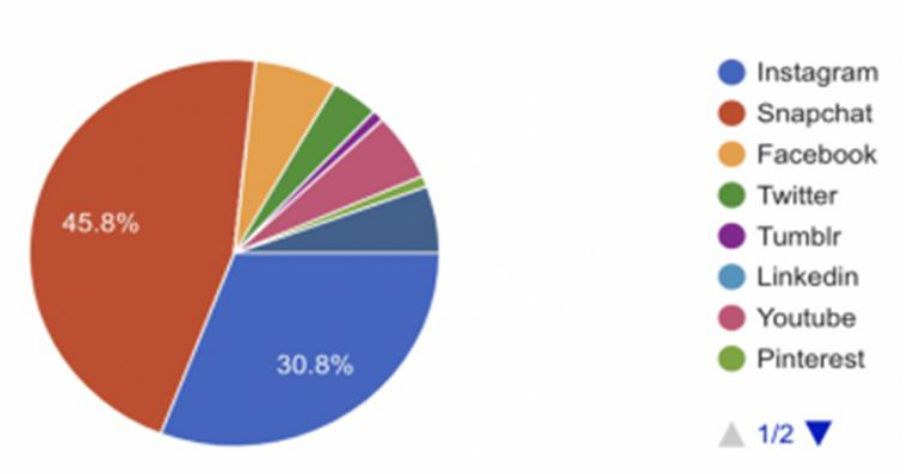


Рисунок 1.2 – Платформи соціальних мереж, що використовувалися у 2022 році [21]

Однією з важливих характеристик задачі є необхідність обробки коротких текстів. На відміну від великих документів, повідомлення та пости в соціальних мережах містять обмежену кількість контексту, що ускладнює визначення настрою

[23, 24]. Методи обробки природної мови мають ефективно використовувати обмежену інформацію та враховувати специфіку коротких повідомлень.

Динамічна природа мови в соціальних різноманітних мережах також створює виклики. Нові слова, сленг та меми постійно з'являються, тому моделі повинні адаптуватися до змін. Статичні словники та моделі швидко застарівають [25–27], що вимагає регулярного оновлення та донавчання на нових даних. Онлайн-навчання та інкрементальні підходи можуть допомогти моделям залишатися актуальними.

Шум у даних соціальних мереж є ще одним важливим фактором. Користувачі часто допускають орфографічні помилки, використовують нестандартну граматику або навмисно змінюють написання слів. Ці особливості вимагають стійких підходів до обробки тексту, які можуть впоратися з варіативністю та помилками. Методи нормалізації тексту та виправлення помилок можуть покращити якість попередньої обробки [28, 29].

Контекстна інформація відіграє важливу роль у визначенні настрою. Одне і теж слово буде мати різне емоційне забарвлення в залежності від контексту. Сучасні моделі обробки природної мови намагаються захопити контекстні залежності через використання рекурентних нейронних мереж або трансформерних архітектур, які можуть моделювати довгострокові залежності в тексті.

Завдання класифікації настроїв тісно пов'язане з іншими задачами обробки природної мови, такими як розпізнавання іменованих сутностей, визначення аспектів та екстракція думок. Аспектно-орієнтована класифікація настроїв фокусується на визначенні настрою відносно конкретних аспектів або об'єктів, згаданих у тексті.

1.2 Аналіз існуючих публікацій та наукових підходів

Аналіз наукової літератури показує розвиток методів класифікації настроїв протягом останніх років. Ранні підходи базувалися на словникових методах, де використовувалися заздалегідь створені словники слів з емоційним забарвленням.

Такі методи були простими у реалізації, але мали обмежену точність через неможливість врахування іншого тексту та складних мовних явищ [1, 4].

Перехід до методів навчання значно покращив якість класифікації настроїв. Класичні алгоритми, такі як баєсівський класифікатор, метод опорних векторів та випадковий ліс, почали активно застосовуватися для цієї задачі. Ці методи вимагають ручного вибору ознак, що включає різноманітні статистичні та лінгвістичні характеристики тексту. Незважаючи на необхідність інженерії ознак, класичні алгоритми показали хороші результати на багатьох датасетах.

Розвиток глибокого навчання призвів до появи нових потужних методів для класифікації настроїв. Рекурентні нейронні мережі стали популярним вибором для обробки послідовних даних, якими є тексти. Довгострокові короткочасні мережі пам'яті дозволяють моделювати довгострокові залежності та захоплювати контекстну інформацію, що важливо для розуміння настрою. Дослідження показують, що двонаправлені довгострокові короткочасні мережі можуть додатково покращити якість за рахунок обробки тексту як з лівого, так і з правого контексту [16, 26, 30].

Згорткові мережі також знайшли область практичного застосування в обробці мови, хоч спочатку вони розроблялися для комп'ютерного зору. Згорткові мережі можуть ефективно виділяти локальні патерни в тексті, такі як біграми та тріграми, що корисно для класифікації настроїв. Комбінація згорткових мереж та довгострокових короткочасних мереж в гібридних архітектурах дозволяє використовувати переваги обох підходів [10].

Революційним моментом у обробці мови стала поява трансформерних архітектур. Модель двонаправленого кодувальника представлень від трансформерів запровадила механізм двонаправленої уваги, що дозволяє кожному слову в тексті взаємодіяти з усіма іншими словами. Це призвело до значного покращення якості на різноманітних задачах обробітку мови, включаючи класифікацію настроїв. Попереднє навчання на великих корпусах тексту дозволяє моделі вивчити загальні лінгвістичні закономірності, які потім можна адаптувати до конкретних задач через тонке налаштування [12–14].

Розширена мовна модель запропонувала альтернативний підхід до моделювання мови через моделювання перестановок мови. Цей метод дозволяє уникнути деяких обмежень попередніх моделей та досягти кращої продуктивності на деяких задачах. Однак висока обчислювальна складність обмежує широке застосування таких моделей [17, 30].

Критичний аналіз публікацій виявляє декілька важливих тенденцій. Спостерігається перехід від класичних методів навчання до глибокого навчання, що забезпечує кращу якість класифікації. Трансформерні архітектури стали домінуючим підходом завдяки своїй ефективності та універсальності. Збільшується увага до багатомовності та міждоменної адаптації моделей [23, 24].

Порівняльний аналіз показує, що вибір методу залежить від специфіки задачі. Для застосувань, де критична швидкість, можуть бути кращим вибором спрощені моделі або словникові методи. Для задач, де важлива максимальна точність, трансформерні архітектури демонструють найкращі результати, хоча вимагають значних обчислювальних ресурсів.

1.3 Огляд архітектур, методів та моделей обробки природної мови для класифікації настроїв

Архітектури обробки природної мови для класифікації настроїв можна поділити на декілька категорій залежно від їх основних принципів роботи. Базові архітектури включають шари векторних представлень, за якими йдуть класифікатори. Вибір архітектури суттєво впливає на якість класифікації та обчислювальну складність моделі [1, 4].

Векторні представлення слів є фундаментальним компонентом більшості моделей. Модель системи Word2Vec також застосовує нейронні мережі для створення зменшених векторних представлень, де семантично схожі слова мають близькі вектори. Модель підтримує два підходи: неперервний мішок слів та скіп-грам. Архітектура скіп-грам часто показує кращі результати для рідкісних слів [5, 22].

Глобальні вектори для представлення слів пропонують альтернативний підхід, базуючись на глобальній статистиці співвустрічаємості слів у корпусі. Ці представлення захоплюють як локальний контекст, так і глобальні статистичні закономірності. Попередньо навчені векторні представлення можна брати як початкові ваги для шару вкладень у нейронній мережі.

Рекурентні мережі спеціально були зроблені для обробки послідовних даних. Базова рекурентна мережа має проблему зникаючого градієнта при обробці довгих послідовностей, що обмежує її здатність захоплювати довгострокові залежності. Архітектура довгострокової короткочасної мережі пам'яті вирішує цю проблему через введення вентилів, що контролюють потік інформації через мережу [16].

Двонаправлені довгострокові короткочасні мережі обробляють текст в обох напрямках, що дозволяє кожному прихованому стану мати інформацію як з минулого, так і з майбутнього контексту. Це особливо корисно для класифікації настроїв, де розуміння повного контексту речення важливе для правильної інтерпретації [13, 15]. Керовані рекурентні блоки представляють спрощену альтернативу довгостроковим короткочасним мережам з меншою кількістю параметрів. Ця архітектура об'єднує вентиль забування та входу в єдиний вентиль оновлення, що робить модель простішою та швидшою у навчанні. На багатьох задачах керовані рекурентні блоки показують порівнянну з довгостроковими мережами якість при меншій обчислювальній складності.

Дистильована версія кодувальника пропонує компактну версію базової моделі з меншою кількістю шарів, що зберігає більшу частину якості оригінальної моделі при значно меншій обчислювальній складності. Дистиляція знань дозволяє навчити меншу модель імітувати поведінку більшої, що корисно для застосувань з обмеженими ресурсами [20, 29].

Полегшена версія кодувальника вводить два ключові покращення: факторизацію матриці вкладень та спільне використання параметрів між шарами. Ці техніки значно зменшують кількість параметрів моделі без суттєвої втрати якості. Для класифікації настроїв це означає можливість використання потужних моделей на

пристроях з обмеженою пам'яттю. Спеціалізовані моделі для соціальних мереж враховують специфіку текстів з Твіттера та інших платформ. Модель для Твіттера попередньо навчена на великій кількості твітів з використанням специфічної токенізації, що краще обробляє хештеги, згадки та емодзі. Така спеціалізація приводить до кращих результатів на задачах аналізу настроїв у соціальних різноманітних мережах порівняно з загальними моделями [15, 16, 26].

Генеративні попередньо навчені трансформери базуються на трансформерному декодері та авторегресивному моделюванні мови. Хоча спочатку розроблялися для генерації тексту, їх також можна адаптувати для класифікації через додавання класифікаційної голови. Великі версії демонструють вражаючі можливості навчання з малою кількістю прикладів. Мультимодальні архітектури інтегрують текстові та візуальні представлення для спільного аналізу. Моделі зору та мови використовують окремі кодувальники для кожної модальності, після чого виконується злиття через конкатенацію, додавання або більш складні механізми уваги. Для класифікації настроїв це дозволяє враховувати емоційний зміст зображень, що супроводжують текст [4, 10].

Гібридні архітектури комбінують переваги різних підходів. Архітектура згорткових мереж та довгострокової пам'яті використовує згорткові шари для виділення локальних ознак, після чого довгострокова мережа обробляє послідовність цих ознак для захоплення глобального контексту. Такі моделі можуть ефективно працювати на задачах класифікації настроїв.

Довгострокова мережа з увагою додає механізм уваги до рекурентної архітектури, що дозволяє моделі фокусуватися на найбільш важливих словах при класифікації. Ваги уваги показують, які частини тексту найбільше вплинули на рішення моделі, що покращує інтерпретованість результатів. Капсульні мережі представляють альтернативний підхід до обробки ієрархічних структур у тексті. Капсули кодують не тільки наявність ознаки, але й її властивості, що може бути корисним для захоплення різних аспектів настрою. Однак висока обчислювальна складність обмежує широке застосування таких архітектур [20, 27].

Графові нейронні мережі дозволяють моделювати структурні залежності в тексті через побудову графів, де вершини представляють слова або фрази, а ребра відображають зв'язки між ними. Для класифікації настроїв графові мережі можуть захоплювати синтаксичні та семантичні зв'язки, що покращує розуміння контексту.

Архітектури з пам'яттю включають зовнішні модулі пам'яті, до яких модель може звертатися під час обробки. Мережі пам'яті дозволяють зберігати та використовувати релевантну інформацію з попередніх прикладів або знань предметної області. Це може бути корисним для обробки складних випадків, що вимагають зовнішніх знань [24, 25].

Змагальне навчання використовується для покращення стійкості моделей. Додавання невеликих збурень до входів під час навчання робить моделі більш стійкими до шуму та атак. Для класифікації настроїв у соціальних різноманітних мережах, де тексти часто містять помилки та нестандартне форматування, така стійкість є важливою. Багатозадачне навчання дозволяє одночасно навчати модель на декількох пов'язаних задачах. Для класифікації настроїв це може включати одночасне визначення настрою та виявлення емоцій, або класифікацію на різних рівнях деталізації. Спільне навчання може покращити якість за рахунок переносу знань між задачами.

Мета-навчання дозволяє моделям швидко адаптуватися до нових задач або доменів з невеликою кількістю прикладів. Алгоритм агностичного до моделі мета-навчання та інші підходи навчають моделі так, щоб вони могли ефективно донавчатися на нових даних. Це особливо корисно для багатомовної класифікації або адаптації до нових соціальних платформ [30].

Техніки навчання з малою кількістю прикладів дозволяють класифікувати настрої в ситуаціях, коли доступно мало анотованих прикладів. Прототипні мережі та мережі порівняння навчають метричний простір, де схожі приклади розташовані близько [31]. Методи на основі підказок використовують можливості великих моделей мови для класифікації через формулювання задачі у вигляді заповнення пропусків. Навчання без прикладів дозволяє класифікувати настрої для категорій, які

не були представлені в навчальних даних. Це досягається через використання семантичних описів класів або переносу знань з пов'язаних задач. Великі попередньо навчені моделі демонструють вражаючі можливості класифікації без прикладів.

Методи адаптації домену вирішують проблему зміщення між навчальним та тестовим доменами. Змагальна адаптація домену навчає модель створювати представлення, інваріантні до домену, що дозволяє краще узагальнювати на нові дані. Для класифікації настроїв це означає можливість навчити модель на одній соціальній платформі та застосувати на іншій [32–34]. Методи пояснюваності та інтерпретованості допомагають зрозуміти, як моделі приймають рішення. Локальна інтерпретована модельно-агностична пояснювальна техніка та адитивні пояснення Шеплі створюють пояснення через апроксимацію складної моделі простішою в локальній області [35–37]. Ваги уваги показують, які слова модель вважає важливими. Ці методи важливі для побудови довіри до систем класифікації настроїв [18, 19].

Змагальне усунення упереджень, перезважування та корекції після обробки допомагають зменшити упередження в моделях класифікації настроїв. Метрики справедливості, такі як демографічна паритетність та вирівняні шанси, використовуються для оцінки та контролю упереджень.

Інкрементальне навчання дозволяє моделям навчатися на нових даних без повного перенавчання. Це важливо для підтримки актуальності моделей класифікації настроїв, оскільки мова та тенденції в соціальних мережах постійно змінюються. Консолідація ваг та інші методи допомагають уникнути катастрофічного забування попередніх знань [22, 27].

Контекстуалізовані вкладення від моделей типу двонаправленого кодувальника надають різні представлення для одного слова в залежності від контексту [38–40]. Це вирішує проблему полісемії, де одне слово може мати різні значення. Для класифікації настроїв контекстуальні представлення дозволяють краще розуміти емоційне забарвлення в складних випадках.

Вкладення речень створюють векторні представлення для цілих речень або документів. Універсальний кодувальник речень, кодувальник речень на основі

трансформерів та інші моделі можуть ефективно кодувати семантику речень. Ці представлення корисні для класифікації настроїв коротких текстів, типових для соціальних мереж [41, 42]. Міжмовні методи дозволяють переносити знання між мовами. Багатомовний кодувальник та крос-мовний кодувальник навчені на текстах багатьма мовами та можуть працювати в режимі без прикладів для класифікації настроїв на мовах, не представлених у навчальних даних. Це особливо цінно для глобальних платформ соціальних мереж [27, 28].

Моделі часової динаміки враховують зміну настроїв у часі. Для соціальних мереж це означає можливість відстежувати еволюцію громадської думки щодо певних тем або подій. Рекурентні архітектури природно підходять для моделювання таких часових залежностей [43].

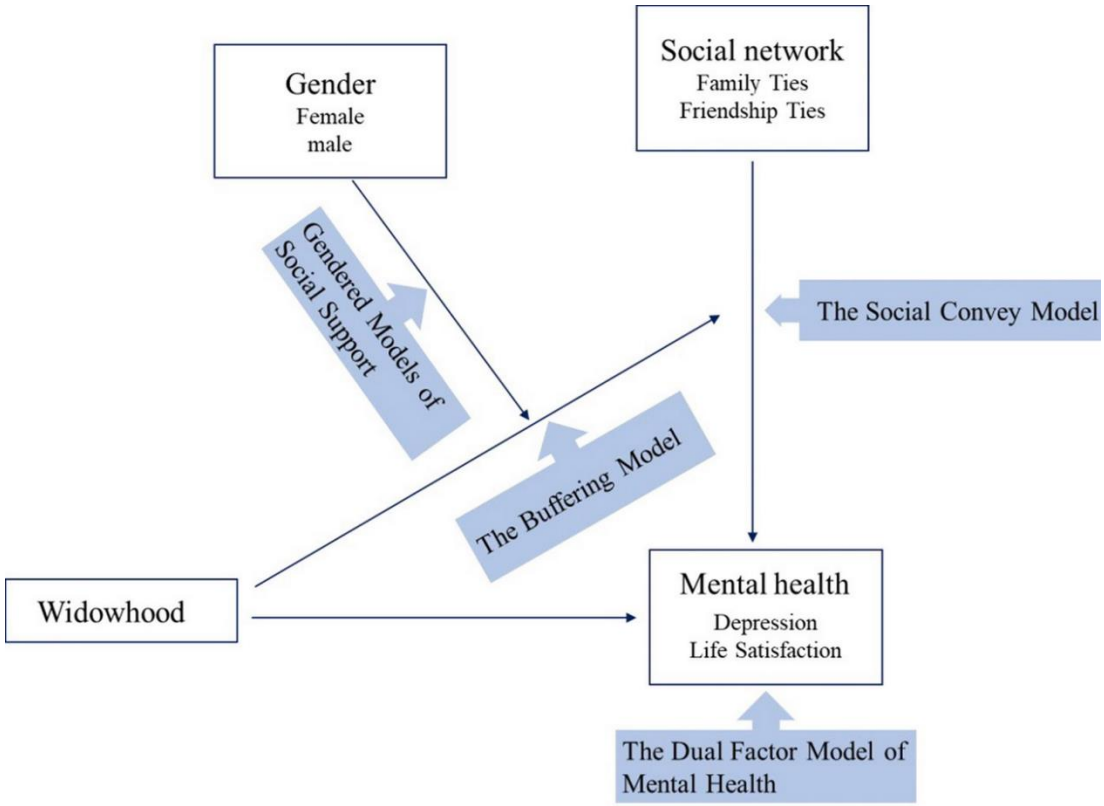


Рисунок 1.3 – Визначення вкладень контексту [38]

Розмовний контекст стає важливим для класифікації настроїв у діалогах або коментарях. Модель повинна враховувати не тільки поточне повідомлення, але й попередній контекст розмови. Ієрархічні архітектури можуть моделювати як рівень окремих повідомлень, так і рівень всієї розмови [44, 45]. Методи перенесення стилю

можуть змінювати емоційне забарвлення тексту, зберігаючи його зміст. Хоча це не безпосередньо класифікація, розуміння того, як змінюється настрій при модифікації тексту, може покращити розуміння факторів, що впливають на емоційний контекст [22].

Підходи контрастного навчання навчають моделі створювати подібні представлення для схожих прикладів та різні для несхожих. Контрастне самоконтрольоване навчання представлень, імпульсне контрастне навчання та інші методи показують хороші результати на різних задачах обробки мови. Для класифікації настроїв контрастивне навчання може покращити якість представлень.

1.4 Мета та постановка задачі

Відповідно до проведеного аналізу сформульовано мету та задачі дослідження.

Мета роботи полягає у підвищенні точності класифікації емоційного забарвлення текстових повідомлень у соціальних мережах шляхом розробки методу з використанням рекурентних нейронних мереж LSTM та спеціалізованої обробки специфічних елементів онлайн-комунікації.

Задачі дослідження:

– провести аналіз існуючих методів та підходів до класифікації настроїв у текстах з використанням методів машинного та глибокого навчання;

– розробити метод класифікації емоційного забарвлення з використанням рекурентних нейронних мереж на основі архітектури LSTM з модифікаціями для обробки емодзі та сленгу;

– здійснити програмну реалізацію методу класифікації емоційного забарвлення текстових повідомлень у соціальних мережах;

– провести експериментальне дослідження ефективності спроектованого методу шляхом порівняння з базовими алгоритмами класифікації та оцінки його точності на тестових даних.

Розділ 2 Метод класифікації настроїв у соціальних мережах та критерії його оцінювання

2.1 Концепція та схема методу класифікації настроїв

Запропонований метод класифікації настроїв у текстах соціальних мереж базується на використанні рекурентних нейронних мереж типу LSTM з додатковим шаром для покращення обробки специфічних елементів соціальних мереж. Основна ідея методу полягає у послідовній обробці текстових даних з урахуванням контекстних залежностей між словами та спеціальними символами.

Концепція методу передбачає декілька ключових етапів обробки вхідних даних. На першому етапі виконується попередня обробка тексту, де відбувається токенізація повідомлень та нормалізація спеціальних символів. Другий етап включає векторизацію токенів з використанням попередньо навчених векторних представлень слів. Третій етап передбачає проходження векторів через рекурентну нейронну мережу для захоплення контекстної інформації. Четвертий етап включає класифікацію з використанням повнозв'язних шарів та функції активації софтмакс.

Метод розроблено для роботи з короткими текстами соціальних мереж, які мають обмежену довжину та часто містять неформальні елементи мови. Вхідні дані представляють собою текстові повідомлення користувачів, а вихідними даними є передбачений клас настрою. У запропонованому підході розглядається три класи настроїв: позитивний, негативний та нейтральний.

Загальна схема методу показана на рисунку 2.1. Схема показує послідовність всіхосновних етапів обробки даних від отримання вхідного тексту до формування остаточного результату класифікації. Кожен етап виконує специфічні функції, що забезпечують поступове перетворення вхідних даних у форму, придатну для класифікації нейронною мережею.

Попередня обробка загального тексту є важливим етапом методу, оскільки якість підготовки даних безпосередньо впливає на точність класифікації. На цьому етапі виконується очищення тексту від зайвих символів, приведення до регістру

малих букв та обробка спеціальних елементів. Токенізація розбиває текст на окремі токени, якими можуть бути слова, емодзі або хештеги.

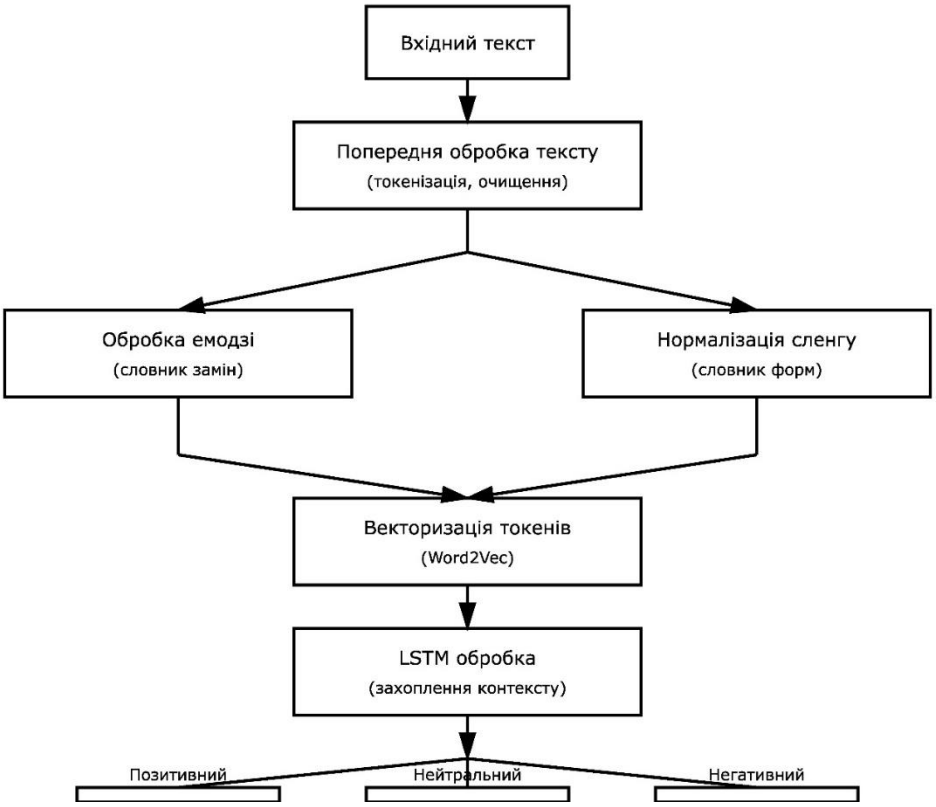


Рисунок 2.1 – Загальна схема методу класифікації настроїв

Векторизація токенів виконується з використанням попередньо навчених векторних представлень Word2Vec або GloVe. Ці представлення дозволяють перетворити текстові токени у числові вектори фіксованої розмірності, де семантично схожі слова мають близькі векторні представлення. Розмірність векторів обирається як компроміс між якістю представлення та обчислювальною складністю.

Рекурентна нейронна мережа LSTM обробляє послідовність векторів токенів з урахуванням їх порядку у тексті. Архітектура LSTM дозволяє захоплювати довгострокові залежності у послідовностях завдяки механізму вентилів, що контролюють потік інформації через мережу. Це особливо важливо для розуміння контексту у коротких повідомленнях соціальних мереж.

Класифікація виконується на основі вихідного представлення від LSTM шару. Повнозв'язні шари перетворюють це представлення у вектор ймовірностей для кожного класу настрою. Функція активації софтмакс забезпечує нормалізацію вихідних значень, так що їх сума дорівнює одиниці, що дозволяє інтерпретувати результати як ймовірності належності до класів.

Метод передбачає навчання на анотованих даних з соціальних мереж, де кожному повідомленню відповідає мітка класу настрою. Процес навчання мінімізує функцію втрат, що вимірює різницю між передбаченими та справжніми мітками класів. Для оптимізації параметрів мережі використовується алгоритм Адам, який адаптує швидкість навчання для кожного параметра.

Особливістю запропонованого методу є врахування специфіки текстів соціальних мереж через спеціальну обробку емодзі та сленгу. Ці елементи несуть важливу емоційну інформацію та потребують окремого підходу до їх представлення та обробки. Модифікація моделі для покращення роботи з такими елементами описана у наступних підрозділах.

2.2 Архітектура моделі класифікації на основі нейронної мережі

Архітектура запропонованої моделі класифікації настроїв складається з декількох послідовно з'єднаних шарів, кожен з яких виконує специфічні функції обробки даних. Базова архітектура побудована на основі рекурентної мережі як LSTM з додатковими шарами для покращення якості класифікації.

Вхідний шар моделі приймає послідовність токенів фіксованої максимальної довжини. Довжина послідовності обмежена значенням 128 токенів, що відповідає типовій довжині повідомлень у соціальних мережах. Коротші послідовності доповнюються спеціальним токеном до максимальної довжини, що забезпечує однаковий розмір вхідних даних для всіх прикладів.

Шар вкладень перетворює токени у векторні представлення розмірністю 300. Цей шар ініціалізується попередньо навченими векторами Word2Vec, що дозволяє

використовувати семантичну інформацію, отриману на великих текстових корпусах. Шар вкладень має параметр, що визначає, чи будуть його ваги оновлюватися під час навчання. У запропонованому підході використовується тонке налаштування вкладень, що дозволяє адаптувати векторні представлення до специфіки задачі класифікації настроїв.

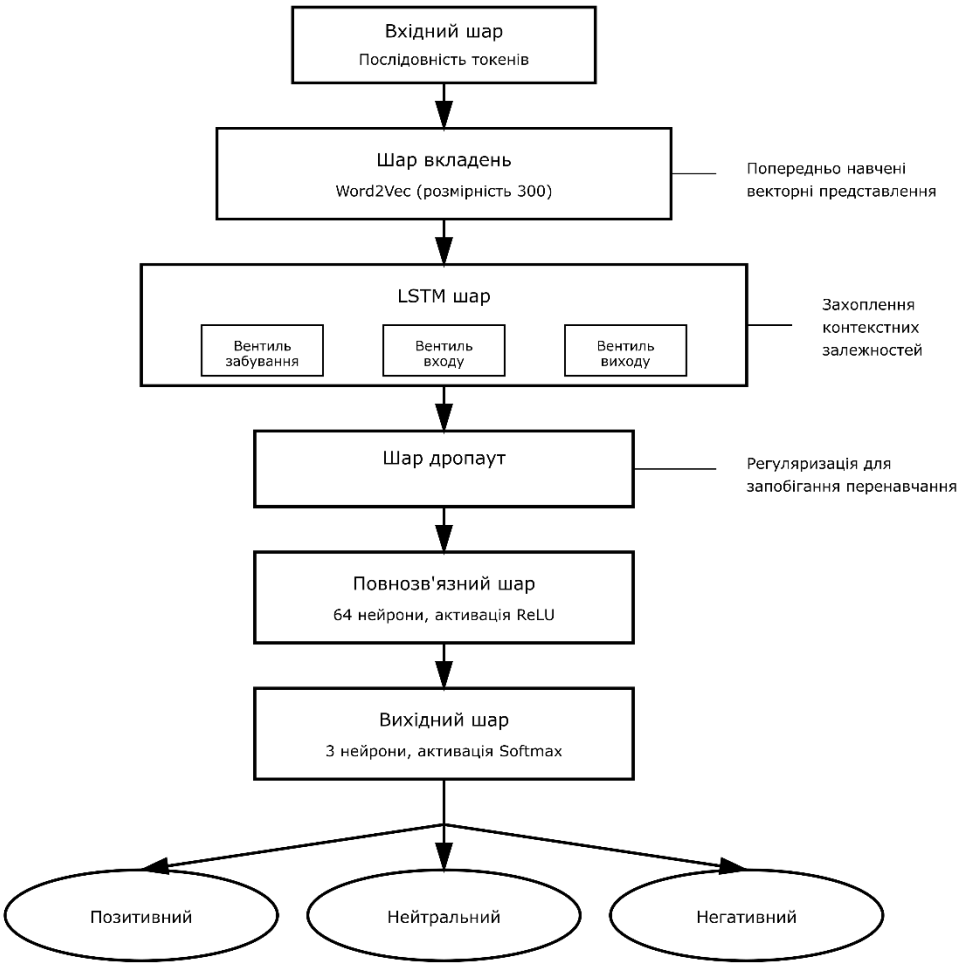


Рисунок 2.2 –Архітектура моделі класифікації настроїв

LSTM шар є основним компонентом моделі, що обробляє послідовність векторних представлень токенів. Цей шар має 128 прихованих одиниць, що забезпечує достатню потужність для захоплення залежностей у текстах соціальних мереж. LSTM використовує три вентилі – вентиль забування, вентиль входу та вентиль виходу – для контролю потоку інформації через мережу.

Вентиль забування визначає, яку частину інформації з попереднього стану необхідно забути. Математично це можна записати як:

$$f_t = \sigma\big(W_f \cdot [h_{t-1}, x_t] + b_f\big) \tag{2.1}$$

де f_t – вектор вентиля забування, σ – сигмоїдна функція активації, W_f – матриця ваг, h_{t-1} – попередній прихований стан, x_t – поточний вхід, b_f – вектор зміщення.

Вентиль входу визначає, яку нову інформацію необхідно додати до стану комірки:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), C = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{2.2}$$

де i_t – вектор вентиля входу, $\tilde{C}_t$ – кандидат на оновлення стану комірки, $tanh$ – гіперболічний тангенс.

Оновлення стану комірки виконується як комбінація забутої частини попереднього стану та нової інформації:

$$C_t = f_t * C_{t-1} + i_t * C \tag{2.3}$$

де C_t – новий стан комірки, * – поелементне множення.

Вентиль виходу визначає, яку частину стану комірки використовувати для формування прихованого стану:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o); \\ h_t = o_t tanh(C_t), \tag{2.4}$$

де o_t – вектор вентиля виходу, h_t – новий прихований стан.

Після LSTM шару розташований шар дропаут з коефіцієнтом 0.5, що використовується для зменшення перенавчання моделі. Дропаут випадково вимикає частину зайвих нейронів під час навчання, що примушує мережу вчитися більш

стійким представленням даних. Під час тестування дропаут не застосовується, і всі нейрони працюють одночасно.

Повнозв'язний шар приймає вихід від LSTM шару та перетворює його у вектор розмірністю 64. Цей шар використовує функцію ReLU, що призводить до нелінійності у модель:

$$z = max(0, W_d \cdot h_{final} + b_d) \quad (2.5)$$

де z – вихід повнозв'язного шару, W_d – матриця ваг, h_{final} – остаточний прихований стан LSTM, b_d – вектор зміщення.

Вихідний шар має три нейрони, що відповідають трьом класам настроїв. Функція активації софтмакс перетворює вихідні значення у розподіл ймовірностей:

$$p_i = exp(z_i)/\Sigma_j exp(z_j) \quad (2.6)$$

де p_i – ймовірність належності до класу i, z_i – вихід для класу i.

Загальна кількість параметрів моделі становить близько 450 тисяч, що є прийнятним значенням для навчання на доступних датасетах. Більшість параметрів зосереджена у шарі вкладень та LSTM шарі, оскільки ці компоненти мають найбільші матриці ваг.

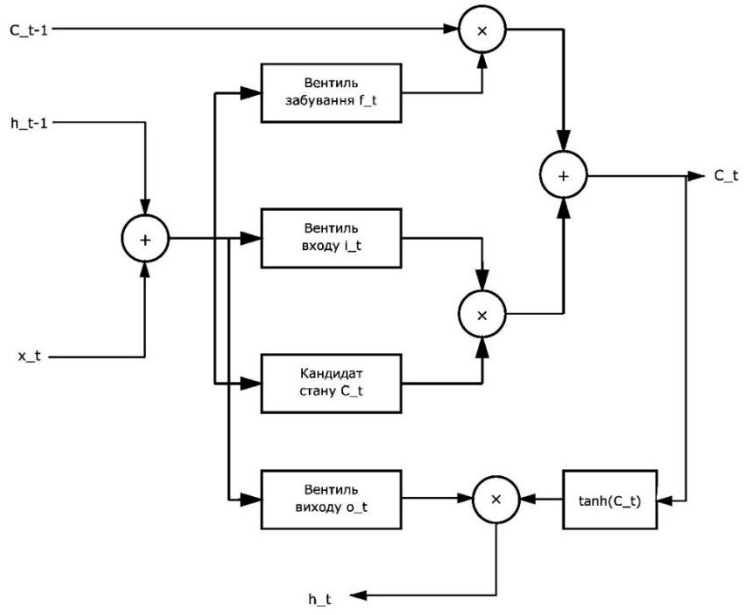


Рисунок 2.3 – Структура LSTM комірки з вентилями

Архітектура моделі спроектована з урахуванням специфіки задачі класифікації настроїв у соціальних мережах. Використання LSTM дозволяє ефективно обробляти послідовності токенів різної довжини та захоплювати контекстні залежності. Шар дропаут запобігає перенавчанню на відносно невеликих датасетах. Повнозв'язні шари забезпечують гнучке відображення прихованих представлень у простір класів настроїв.

Модель реалізована з використанням бібліотеки Keras, що спрощує процес визначення архітектури та навчання. Використання високорівневих бібліотек дозволяє зосередитися на розробці ефективної архітектури, а не на деталях реалізації операцій нейронної мережі.

2.3 Модифікація моделі для покращення обробки емодзі та сленгу

Тексти соціальних мереж містять значну кількість емодзі та сленгових виразів, що несуть важливу інформацію про настрій автора повідомлення. Стандартні підходи до обробки тексту часто не враховують специфіку цих елементів, що призводить до втрати корисної інформації для класифікації. Запропонована модифікація моделі спрямована на покращення обробки емодзі та сленгу через додавання спеціального шару попередньої обробки.

Основна ідея модифікації полягає у створенні словника емодзі з відповідними їм текстовими описами емоцій. Кожен емодзі замінюється на текстовий опис, що відображає його емоційне значення. Наприклад, емодзі усміхненого обличчя замінюється на слово "щасливий", а емодзі сумного обличчя на слово "сумний". Цей підхід дозволяє використовувати наявні векторні представлення слів для кодування інформації про емодзі.

Словник емодзі містить найбільш поширені емодзі, що зустрічаються у датасеті соціальних мереж. Для кожного емодзі визначається основна емоція або настрій, який він передає. Словник створений на основі аналізу використання емодзі у анотованих повідомленнях та узагальнення їх емоційного значення.

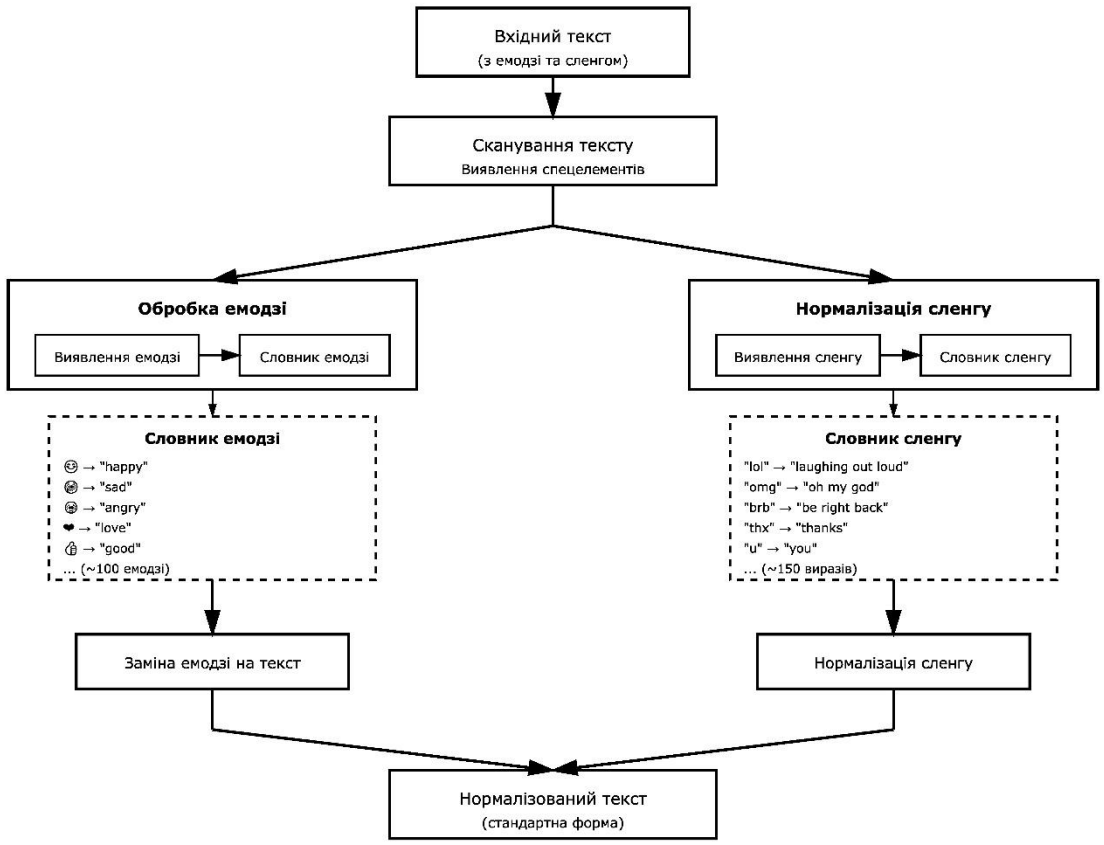


Рисунок 2.4 – Схема обробки емодзі та сленгу у модифікованій моделі

Обробка сленгових виразів виконується аналогічним чином через створення словника сленгу з відповідними нормалізованими формами. Сленгові вирази часто є скороченими або неформальними версіями стандартних слів та фраз. Нормалізація перетворює сленг у стандартні форми, що покращує розпізнавання їх значення моделлю.

Словник сленгу містить типові скорочення та неформальні вирази англійської мови, що часто зустрічаються у соціальних мережах. Наприклад, вираз "lol" нормалізується до "laughing out loud", а "omg" до "oh my god". Створення словника виконано через аналіз навчального датасету та виділення найбільш частотних сленгових форм.

Модифікація моделі включає додавання етапу попередньої обробки перед шаром вкладень. На цьому етапі виконується заміна емодзі та сленгу відповідно до

створених словників. Послідовність обробки включає спочатку заміну емодзі, а потім нормалізацію сленгу, що забезпечує правильне перетворення вхідного тексту.

Алгоритм обробки емодзі працює наступним чином. Спочатку виконується сканування вхідного тексту для виявлення символів емодзі. Кожен знайдений емодзі перевіряється на наявність у словнику. Якщо емодзі присутній у словнику, він замінюється на відповідний текстовий опис. Якщо емодзі відсутній у словнику, він видаляється з тексту, оскільки його значення невідоме моделі.

Алгоритм нормалізації сленгу виконує пошук сленгових виразів у тексті та їх заміну на стандартні форми. Пошук виконується через перевірку кожного токена на наявність у словнику сленгу. Для врахування можливих варіацій написання сленгових виразів використовується нечутливе до регістру порівняння.

Ефективність модифікації оцінюється через порівняння якості класифікації базової та модифікованої моделей на тестовому датасеті. Метрики точності, повноти та F1-оцінки використовуються для кількісного оцінювання покращення. Окремо аналізується якість класифікації повідомлень, що містять емодзі та сленг, для виявлення впливу модифікації на обробку цих елементів.

Приклад обробки тексту з емодзі та сленгом демонструє роботу модифікації. Вхідний текст "This movie is so cool 😊 lol" перетворюється на "This movie is so cool happy laughing out loud" після застосування словників. Такий текст краще обробляється моделлю, оскільки всі елементи представлені у стандартній текстовій формі.

Модифікація не вимагає змін у архітектурі нейронної мережі, а лише додає етап попередньої обробки даних. Це робить підхід простим у реалізації та інтеграції у існуючі системи класифікації настроїв. Створені словники можуть легко розширюватися для додавання нових емодзі та сленгових виразів у міру їх появи у соціальних мережах.

Обмеженням запропонованої модифікації є залежність від якості та повноти словників. Емодзі та сленг, що не представлені у словниках, не обробляються

належним чином. Для покращення покриття необхідне регулярне оновлення словників на основі аналізу нових даних з соціальних мереж.

2.4 Формування та підготовка навчальних даних

Якість навчальних даних значно визначає коректність навчання моделі класифікації настроїв. Формування датасету включає збір текстових повідомлень з соціальних мереж та їх анотування відповідно до класів настроїв. У запропонованому методі використовується публічно доступний датасет з платформи Kaggle, що містить повідомлення з різних соціальних мереж з мітками настроїв.

Датасет містить повідомлення, розділені на три класи: позитивний настрій, негативний настрій та нейтральний настрій. Розподіл прикладів між класами є відносно збалансованим, що спрощує процес навчання моделі. Кожне повідомлення супроводжується міткою класу, що дозволяє використовувати контрольоване навчання.

Структура датасету включає два основні поля: текст повідомлення та мітка класу настрою. Текст повідомлення представлений у вигляді рядка символів, що може містити букви, цифри, спеціальні символи, емодзі та хештеги. Мітка класу кодується цілим числом, де 0 відповідає негативному настрою, 1 нейтральному, а 2 позитивному.

Попередня обробка даних є критичним етапом підготовки датасету. Цей процес включає декілька послідовних кроків очищення та нормалізації тексту. Перший крок передбачає видалення URL-адрес з повідомлень, оскільки вони не несуть інформації про настрій. Регулярні вирази використовуються для виявлення та видалення посилань.

Другий крок включає видалення згадок користувачів, що позначаються символом "@" з наступним іменем користувача. Згадки також не містять інформації про настрій повідомлення та можуть заважати навчанню моделі. Аналогічно до URL-адрес, згадки видаляються з використанням регулярних виразів.

Третій крок передбачає обробку хештегів. На відміну від URL-адрес та згадок, хештеги можуть містити корисну інформацію про настрій. Тому символ "#" видаляється, але текст хештегу зберігається. Наприклад, хештег "#happy" перетворюється на слово "happy".

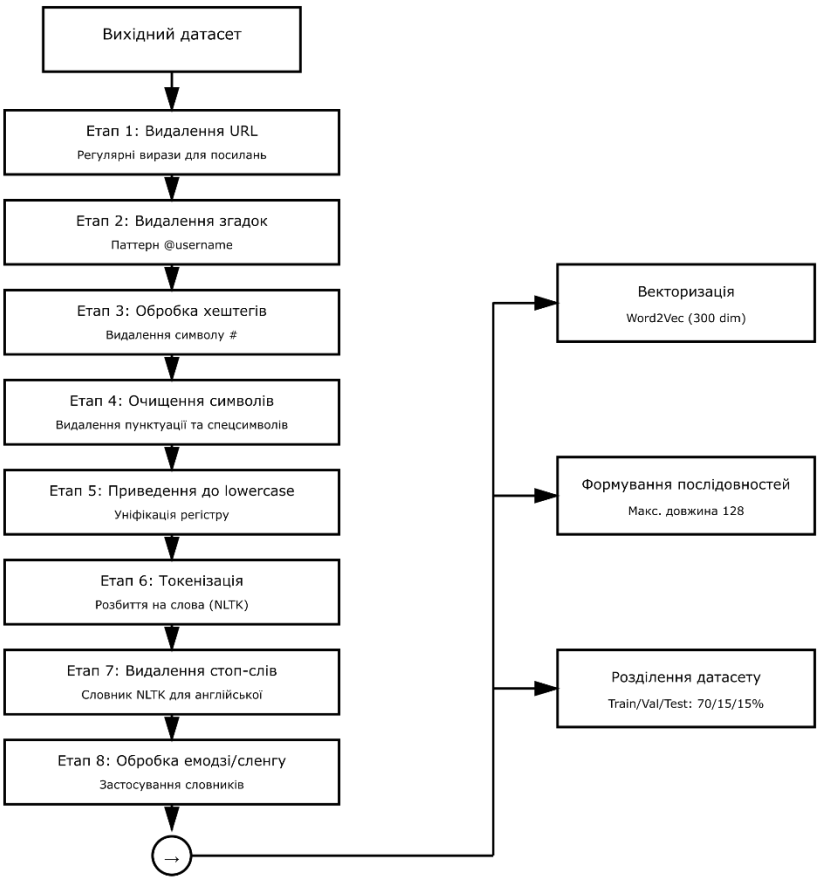


Рисунок 2.5 – Етапи підготовки навчальних даних

Четвертий крок включає видалення спеціальних символів та знаків пунктуації. Зберігаються лише букви, цифри, пробіли та емодзі. Цей крок спрощує текст та зменшує розмірність словника токенів. Регулярні вирази використовуються для ідентифікації та видалення небажаних символів.

П'ятий крок передбачає приведення всього тексту до нижнього регістру. Це робиться для уніфікації представлення слів, оскільки модель повинна розглядати слова "Happy" та "happy" як однакові токени. Приведення до нижнього регістру виконується через стандартні функції обробки рядків.

Шостий крок включає токенізацію тексту на окремі слова та символи. Токенізація виконується з використанням бібліотеки NLTK, що забезпечує правильне розбиття тексту з урахуванням особливостей англійської мови. Результатом токенізації є послідовність токенів для кожного повідомлення.

Сьомий крок передбачає видалення стоп-слів – часто вживаних слів, що не несуть значної семантичної інформації. До стоп-слів відносяться такі слова як "the", "is", "at", "of" тощо. Список стоп-слів береться з бібліотеки NLTK для англійської мови. Видалення стоп-слів зменшує розмірність даних та допомагає моделі зосередитися на більш значущих словах.

Восьмий крок включає застосування модифікації для обробки емодзі та сленгу. Словники емодзі та сленгу використовуються для перетворення цих елементів у стандартну текстову форму. Цей крок виконується після базового очищення тексту, але перед векторизацією.

Векторизація токенів виконується з використанням попередньо навчених векторів Word2Vec. Завантажуються векторні представлення розмірністю 300, навчені на великому корпусі текстів Google News. Для кожного токена у датасеті знаходиться відповідний вектор. Токени, що відсутні у Word2Vec, замінюються на нульовий вектор.

Формування послідовностей фіксованої довжини є наступним кроком підготовки даних. Встановлюється максимальна довжина послідовності у 128 токенів, що відповідає 95-му перцентилю довжин повідомлень у датасеті. Послідовності, що перевищують цю довжину, обрізаються, а коротші послідовності доповнюються спеціальним токеном паддінгу до максимальної довжини.

Навчальна вибірка використовується для перепису вагових значень моделі під час поточного навчання. Валідаційна вибірка застосовується для налаштування гіперпараметрів та контролю процесу навчання. Тестова вибірка призначена для остаточного оцінювання якості навченої моделі.

Балансування класів у навчальній вибірці є важливим аспектом підготовки даних. Хоча датасет є відносно збалансованим, невеликі відмінності у кількості

прикладів різних класів можуть призвести до зміщення моделі. Для компенсації дисбалансу використовуються ваги класів під час поточного навчання, що надають більшу вагу помилкам на менш представлених класах.

Аугментація даних розглядається як можливий спосіб розширення навчального датасету. Однак для задачі класифікації настроїв аугментація складніша, ніж для задач комп'ютерного зору, оскільки необхідно зберігати семантичне значення та настрій тексту. Прості методи аугментації, такі як заміна слів синонімами, можуть змінити настрій повідомлення. Тому у запропонованому методі аугментація не застосовується.

Статистика датасету після попередньої обробки показує, що середня довжина послідовності становить 42 токени, а словник містить близько 25000 унікальних токенів. Розподіл класів: позитивний настрій – 38%, негативний настрій – 33%, нейтральний настрій – 29%. Такий розподіл є достатньо збалансованим для навчання моделі без застосування спеціальних технік балансування.

2.5 Критерії та метрики оцінювання роботи методу класифікації

Оцінювання якості роботи методу класифікації настроїв виконується з використанням набору метрик, що характеризують різні аспекти продуктивності моделі. Вибір метрик обумовлений специфікою задачі багатокласової класифікації та необхідністю збалансованого оцінювання якості для всіх класів настроїв.

Основною метрикою є точність класифікації, що визначається як відношення кількості правильно в поточному значені класифікованих прикладів до загальної кількості прикладів:

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \quad (2.7)$$

де TP – кількість істинно позитивних прикладів, TN – істинно негативних, FP – хибно позитивних, FN – хибно негативних.

Точність дає загальне уявлення про якість класифікації, але може бути оманливою на незбалансованих даних. Для багатокласової задачі точність розраховується як середнє значення по всіх класах або як загальна точність для всього датасету.

Прецизійність вимірює частку правильно детекованих прикладів певного класу серед всіх прикладів, класифікованих як цей клас:

$$Precision = TP/(TP + FP) \tag{2.8}$$

Висока прецизійність означає, що модель рідко помилково відносить приклади інших класів до даного класу. Для багатокласової задачі прецизійність розраховується окремо для кожного класу.

Повнота показує, яку частину прикладів класу модель змогла правильно визначити:

$$Recall = TP/(TP + FN) \tag{2.9}$$

Висока повнота означає, що модель знаходить більшість прикладів даного класу. Для багатокласової задачі повнота також розраховується окремо для кожного класу настрою.

F1-оцінка є гармонійним середнім між прецизійністю та повнотою. Ця метрика особливо корисна, коли необхідно збалансувати прецизійність та повноту. F1-оцінка досягає максимального значення 1, коли прецизійність та повнота обидві дорівнюють 1, і мінімального значення 0, коли одна з них дорівнює 0.

Для багатокласової задачі F1-оцінка розраховується двома способами. Макро-усереднена F1-оцінка обчислюється як середнє арифметичне F1-оцінок для всіх класів:

Зважена F1-оцінка враховує кількість прикладів кожного класу, де w_i – вага класу i, що дорівнює частці прикладів цього класу у датасеті, $F1_i$ – F1-оцінка для

класу i. Матриця помилок є корисним інструментом для візуалізації та аналізу якості класифікації. Для трикласової задачі матриця помилок має розмір 3×3, де елемент на позиції (i, j) показує кількість прикладів класу i, класифікованих як клас j. Діагональні елементи відповідають правильно класифікованим прикладам, а недіагональні – помилкам. Аналіз матриці помилок дозволяє виявити, між якими класами модель найчастіше плутається. Наприклад, якщо модель часто класифікує нейтральні повідомлення як позитивні, це видно з відповідного елемента матриці. Така інформація корисна для подальшого вдосконалення моделі.

Крім основних метрик класифікації, важливими є також метрики продуктивності моделі. Час навчання вимірює тривалість процесу навчання моделі на навчальному датасеті. Цей параметр важливий для оцінки практичності методу, оскільки занадто довге навчання може бути неприйнятним для практичного застосування.

Час необхідний для класифікації одного повідомлення навченою моделлю. Для застосувань реального часу важливо, щоб цей час був мінімальним. Вимірювання виконується на тестовому датасеті з розрахунком середнього часу обробки одного прикладу.

Використання пам'яті є ще одним важливим параметром, особливо для розгортання моделі на різноманітних девайсах з обмеженими ресурсами. Вимірюється як обсяг оперативної пам'яті, необхідний для завантаження та роботи моделі. Модель з меншою кількістю параметрів вимагає менше пам'яті.

Розмір моделі визначається кількістю параметрів та їх типом. Для запропонованої архітектури розмір моделі становить близько 450 тисяч параметрів, що відповідає приблизно 1.8 МБ при збереженні у форматі з плаваючою комою одинарної точності. Це є прийнятним розміром для більшості практичних застосувань. Крива навчання показує зміну метрик якості на навчальній та валідаційній вибірках у процесі поточного навчання. Аналіз кривої навчання дозволяє виявити перенавчання моделі, яке проявляється у розбіжності між якістю на навчальній та валідаційній вибірках. Якщо модель перенавчається, якість на

навчальній вибірці продовжує покращуватися, тоді як на валідаційній починає погіршуватися.

Крива точності показує зміну accuracy на навчальній та валідаційній вибірках по епохах навчання. Ідеальна крива характеризується монотонним зростанням обох значень до певного рівня насичення. Велика різниця між кривими свідчить про перенавчання.

Крива функції втрат демонструє зміну значення функції втрат під час навчання. Функція втрат для задачі класифікації зазвичай визначається як категоріальна крос-енентропія. Зменшення функції втрат на навчальній вибірці свідчить про процес навчання моделі. Якщо втрати на валідаційній вибірці починають зростати при продовженні зменшення втрат на навчальній вибірці, це сигналізує про перенавчання. Порівняння з базовими моделями є важливим аспектом оцінювання запропонованого методу. Базові моделі включають прості підходи до класифікації, такі як баєсівський класифікатор та інші . Порівняння виконується на одному і тому самому тестовому датасеті з використанням однакових метрик.

Наївний баєсівський класифікатор використовується як простий baseline, що базується на імовірнісному підході. Цей метод обчислює ймовірність належності до класу на основі частотного аналізу слів у навчальному датасеті. Незважаючи на простоту, наївний баєс часто показує прийнятні результати на задачах класифікації текстів. Логістична регресія є лінійним методом класифікації, що може використовуватися для багатокласових задач через підхід один-проти-всіх. Модель навчається знаходити лінійну межу між класами у просторі ознак. Для представлення тексту використовується TF-IDF векторизація.

Метод опорних векторів є більш складним підходом, що будує нелінійну межу рішення через використання ядрової функції. SVM часто демонструє хороші результати на задачах класифікації тексту, особливо на збалансованих датасетах.

Крос-валідація використовується для більш надійного оцінювання якості моделі. Датасет розділяється на k частин, і навчання виконується k разів, кожного разу використовуючи одну частину для тестування, а решту для навчання. Результати

усереднюються для отримання оцінки якості, що менш залежить від конкретного розбиття даних. Аналіз помилок класифікації допомагає виявити слабкі місця моделі. Виконується детальний розгляд прикладів, що були неправильно класифіковані, для розуміння причин помилок. Типові причини включають сарказм, складні граматичні конструкції, змішані настрої у одному повідомленні та неоднозначний контекст.

Чутливість до гіперпараметрів досліджується через експерименти з різними значеннями ключових параметрів моделі. Основні гіперпараметри включають розмір прихованого шару LSTM, коефіцієнт дропауту, швидкість навчання та розмір батчу. Для кожного параметра визначається діапазон значень, у якому якість моделі залишається стабільною.

Узагальнююча здатність моделі оцінюється через тестування на даних з інших джерел. Хоча модель навчена на конкретному датасеті, бажано, щоб вона могла класифікувати повідомлення з інших соціальних мереж або тематик. Це характеризує стійкість моделі та можливість її застосування у різних контекстах.

Інтерпретованість результатів є важливим аспектом для практичного застосування моделі. Користувачам системи корисно розуміти, чому модель прийняла певне рішення. Для цього можуть використовуватися техніки візуалізації уваги або аналізу впливу окремих слів на класифікацію.

Висновки до розділу 2

У другому розділі описано теоретичні основи запропонованого методу класифікації настроїв у текстах соціальних мереж з використанням обробки мови. Розроблено концепцію методу, що базується на послідовній обробці текстових даних через етапи попередньої обробки, векторизації, обробки рекурентною нейронною мережею та класифікації.

Запропонована архітектура моделі включає шар вкладень з попередньо навченими векторами Word2Vec, LSTM шар для захоплення контекстних залежностей, шар дропаут для зменшення перенавчання та повнозв'язні шари для

класифікації. Архітектура спроектована з урахуванням специфіки обмежених текстів соціальних мереж та забезпечує ефективну обробку послідовностей токенів.

Розроблено модифікацію моделі для покращення обробки емодзі та сленгових виразів, що часто зустрічаються у текстах соціальних мереж. Модифікація включає створення словникових масиві емодзі та сленгу з відповідними текстовими описами та нормалізованими формами. Застосування словників виконується на етапі попередньої обробки даних, що дозволяє поєднувати модифікацію без змін у архітектурі нейронної мережі.

Описано процес формування та підготовки навчальних даних, що включає попередню обробку тексту, токенізацію, видалення стоп-слів, векторизацію та формування послідовностей фіксованої довжини.

Визначено критерії та метрики оцінювання роботи методу класифікації, що включають точність, прецизійність, повноту, F1-оцінку та матрицю помилок. Метрики розраховуються окремо для кожного класу настрою, що дозволяє виявити специфіку класифікації різних типів настроїв. Додатково враховуються метрики продуктивності моделі, такі як час навчання, час інференсу та використання пам'яті.

Описано підходи до порівняння запропонованого методу з базовими моделями, включаючи баєсівський класифікатор, логістичну регресію та метод опорних векторів. Порівняння виконується на одному тестовому датасеті з використанням однакових метрик для забезпечення об'єктивності оцінювання.

Запропонований метод має практичну цінність для задач аналізу настроїв у соціальних мережах завдяки врахуванню специфіки таких текстів через модифікацію обробки емодзі та сленгу. Архітектура моделі є достатньо простою для реалізації та навчання на доступних обчислювальних ресурсах, що робить метод придатним для практичного застосування.

Розділ 3 Програмна реалізація методу класифікації настроїв у соціальних мережах

3.1 Вибір технологічного стеку та інструментарію розробки

Реалізація методу класифікації настроїв потребує ретельного вибору програмних засобів та бібліотек, що забезпечать надійну роботу системи. Основним критерієм вибору технологій стала їх широка підтримка у спільноті розробників, наявність документації та можливість інтеграції з існуючими рішеннями у галузі обробки природної мови.

Python версії 3.10 обрано як основну мову програмування. Цей вибір обумовлений кількома факторами. Python має розвинену екосистему бібліотек для машинного навчання та обробки текстів. Синтаксис мови дозволяє швидко прототипувати рішення та проводити експерименти з різними підходами. Підтримка роботи з великими обсягами даних забезпечується оптимізованими бібліотеками нижнього рівня.

Для побудови та навчання нейронної мережі використовується бібліотека TensorFlow 2.13 разом з високорівневим API Keras. TensorFlow забезпечує гнучкість у визначенні архітектури моделі та можливість масштабування обчислень. Keras спрощує процес створення шарів мережі та управління процесом навчання. Інтеграція між цими інструментами дозволяє поєднувати простоту використання з потужними можливостями налаштування.

Обробка текстових даних виконується за допомогою бібліотеки NLTK версії 3.8. Ця бібліотека надає широкий набір функцій для токенізації, нормалізації та попередньої обробки текстів. NLTK містить готові списки стоп-слів для англійської мови, що використовуються на етапі очищення даних. Функціонал бібліотеки дозволяє гнучко налаштовувати процес обробки під специфіку текстів соціальних мереж.

Для роботи з векторними представленнями слів застосовується бібліотека Gensim 4.3. Вона підтримує завантаження попередньо навчених моделей Word2Vec

та надає інструменти для тонкого налаштування векторів. Gensim оптимізована для роботи з великими корпусами текстів та забезпечує швидкий доступ до векторних представлень.

NumPy версії 1.24 використовується для операцій з числовими масивами та матрицями. Ця бібліотека є основою для багатьох інших інструментів машинного навчання. NumPy забезпечує швидкі векторизовані операції над даними, що критично важливо при обробці великих датасетів.

Pandas 2.0 застосовується для завантаження, очищення та трансформації табличних даних. Бібліотека надає необхідні структурні дані DataFrame для роботи з датасетом. Функції Pandas дозволяють легко виконувати фільтрацію, групування та агрегацію даних на етапі підготовки.

Візуалізація результатів та метрик навчання реалізована за допомогою використання Matplotlib 3.7 та Seaborn 0.12. Matplotlib надає базові можливості побудови графіків та діаграм. Seaborn додає стилізацію та спеціалізовані типи візуалізацій для статистичних даних. Обидві бібліотеки інтегруються між собою та дозволяють створювати наочні представлення результатів.

Для оцінювання якості моделі використовуються інструменти з scikit-learn 1.3. Ця бібліотека містить реалізації метрик класифікації, функції для розділення датасету та інструменти для поперемінної валідації. Scikit-learn забезпечує стандартизований інтерфейс для роботи з різними моделями машинного навчання.

Середовище розробки дозволяє поєднувати код, візуалізації та текстові пояснення в одному інтерактивному документі. Система забезпечує зручність експериментування з різними підходами та швидке прототипування рішень. Можливість виконання коду блоками спрощує налагодження та тестування окремих компонентів системи.

Система для контролювання версій Git використовується для відстежування змін у коді та управління різними версіями моделі. Репозиторій розміщено на платформі GitHub, що забезпечує зберігання коду та можливість співпраці. Структура

репозиторію організована відповідно до загальноприйнятих практик, з окремими директоріями для коду, даних та документації.

Для створення словників емодзі та сленгу використовується бібліотека emoji 2.8, що надає інструменти для виявлення та обробки емодзі у текстах. Словник сленгу формується вручну на основі аналізу найчастотніших виразів у навчальному датасеті.

Обрані технології забезпечують повний цикл розробки від попередньої обробки даних до оцінювання навченої моделі. Сумісність між бібліотеками дозволяє легко інтегрувати різні компоненти системи.

3.2 Структура програмного рішення та основні компоненти

Програмна реалізація методу організована у вигляді модульної системи з чітким розділенням відповідальності між компонентами. Така архітектура забезпечує можливість незалежної розробки та тестування окремих модулів, а також спрощує подальше розширення функціоналу.

Структура проєкту включає кілька основних директорій. Директорія data містить вхідні датасети та проміжні результати обробки. Папка src зберігає весь програмний код, організований за модулями. Тека model_z призначена для зберігання навчених моделей та їх параметрів. Папка notebooks містить Jupyter-ноутбуки для експериментів та аналізу. Директорія outputs зберігає результати експериментів, графіки та таблиці метрик.

Модуль препроцесингу data_preprocessing.py відповідає за завантаження та початкову обробку текстових даних. Основний клас TextPreprocessor інкапсулює всю логіку очищення текстів. Метод load_dataset завантажує дані з CSV-файлу та виконує базову валідацію. Функція clean_text реалізує послідовне застосування операцій очищення до кожного повідомлення.

Очищення тексту включає кілька послідовних кроків. Спочатку видаляються URL-адреси за допомогою регулярного виразу, що ідентифікує патерни веб-посилань. Далі видаляються згадки користувачів, що починаються з символу @.

Хештеги обробляються шляхом видалення символу # із збереженням тексту. Спеціальні символи та знаки пунктуації фільтруються, залишаючи лише букви, цифри та пробіли.

Модуль emoji_handler.py реалізує обробку емодзі у текстах. Клас EmojiProcessor містить словник відповідностей між емодзі та їх текстовими описами. Метод process_emojis сканує текст на наявність символів емодзі та замінює їх відповідними словами. Словник емодзі включає найпоширеніші символи, що зустрічаються у датасеті соціальних мереж.

Для обробки сленгу створено модуль slang_normalizer.py. Клас SlangNormalizer містить словник сленгових виразів та їх нормалізованих форм. Метод normalize_slang перевіряє кожен токен тексту на наявність у словнику та виконує заміну при знаходженні збігу. Словник формується на основі аналізу частотності термінів у навчальних даних.

Токенізація та лематизація реалізовані у модулі text_tokenizer.py. Клас TextTokenizer використовує можливості NLTK для розбиття тексту на окремі слова. Метод tokenize застосовує word_tokenize з NLTK для поділу речень. Функція remove_stopwords фільтрує службові слова на основі списку стоп-слів для англійської мови.

Векторизація текстів виконується модулем embedding_manager.py. Клас EmbeddingManager завантажує попередньо навчені вектори Word2Vec та забезпечує доступ до них. Метод load_word2vec_model читає бінарний файл з векторними представленнями. Функція vectorize_tokens перетворює послідовність токенів у матрицю векторів фіксованої розмірності.

Для формування послідовностей фіксованої довжини створено модуль sequence_processor.py. Клас SequenceProcessor реалізує паддінг коротких послідовностей та обрізання довгих. Метод pad_sequences доповнює послідовності нульовими векторами до максимальної довжини. Функція truncate_sequences обмежує довжину послідовностей заданим значенням.

Архітектура нейронної мережі визначена у модулі model_architecture.py. Клас SentimentLSTMModel інкапсулює побудову всіх шарів моделі. Метод build_model створює послідовну архітектуру з використанням Keras Sequential API. Функція add_embedding_layer додає шар вкладень з ініціалізацією з Word2Vec. Метод add_lstm_layer конфігурує рекурентний шар з вказаною кількістю прихованих одиниць.

Процес навчання моделі організовано у модулі model_trainer.py. Клас ModelTrainer керує циклом навчання та валідації. Метод compile_model налаштовує функцію втрат, оптимізатор та метрики. Функція fit_model запускає процес навчання з вказаними параметрами. Метод save_model зберігає навчену модель у файл для подальшого використання.

Модуль evaluation.py відповідає за оцінювання якості навченої моделі. Клас ModelEvaluator обчислює різні метрики класифікації на тестовому датасеті. Метод calculate_metrics розраховує точність, прецизійність, повноту та F1-оцінку для кожного класу. Функція generate_confusion_matrix будує матрицю помилок для аналізу типових помилок моделі.

Візуалізація результатів реалізована у модулі visualization.py. Клас ResultsVisualizer створює графіки та діаграми для представлення метрик. Метод plot_training_history відображає криві навчання для точності та функції втрат. Функція plot_confusion_matrix створює теплову карту матриці помилок. Метод plot_metrics_comparison будує стовпчасті діаграми для порівняння метрик різних моделей.

Головний модуль main.py об'єднує всі компоненти системи та організовує їх взаємодію. Функція main визначає послідовність виконання етапів від завантаження даних до збереження результатів. Цей модуль читає конфігураційні параметри, ініціалізує необхідні об'єкти та керує потоком виконання програми.

Конфігураційні параметри зберігаються у файлі config.py. Тут визначені константи для шляхів до файлів, параметрів моделі та налаштувань навчання.

Централізоване зберігання конфігурації спрощує внесення змін та експериментування з різними параметрами.

Для обробки помилок створено модуль exceptions з визначенням власних класів винятків. Це дозволяє точніше ідентифікувати проблеми на різних етапах роботи системи та надавати інформативні повідомлення про помилки.

Модуль utils.py містить допоміжні функції загального призначення. Сюди входять функції для безпосередньої роботи з файлами, форматування даних та інші утилітарні операції, що використовуються у різних частинах системи.

Тестування компонентів організовано у директорії tests. Для кожного основного модуля створено відповідний тестовий файл з набором unit-тестів. Використовується фреймворк pytest для автоматизації запуску тестів та перевірки коректності роботи функцій. На рисунку 3.1 показано діаграму компонентів системи класифікації настроїв.

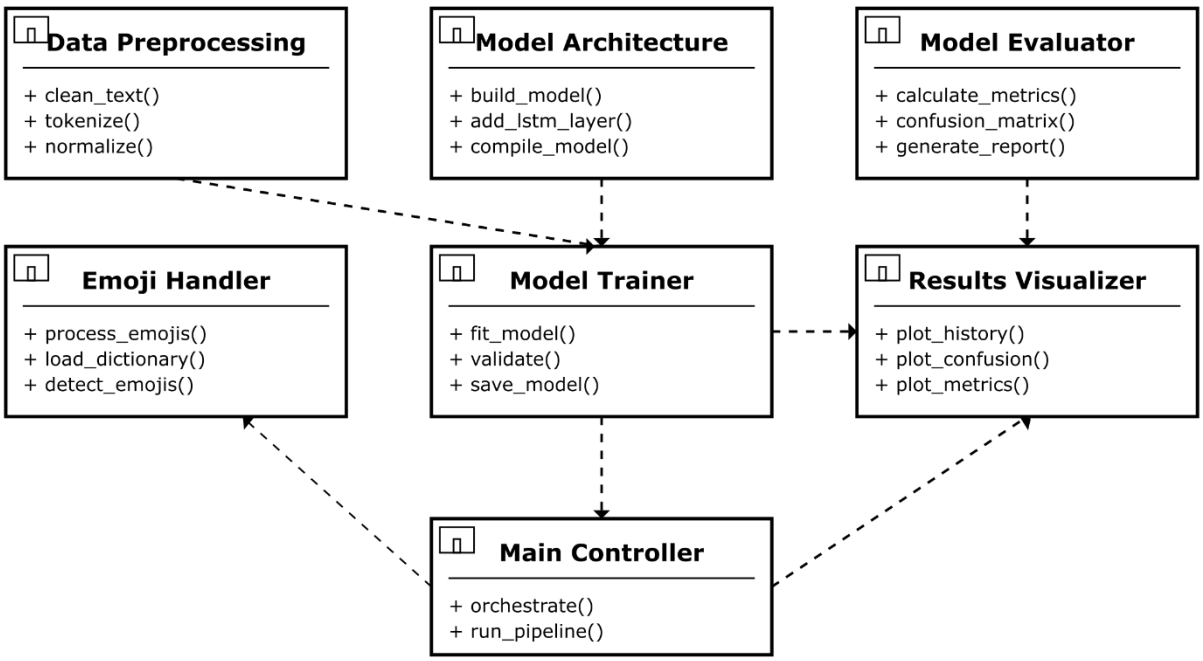


Рисунок 3.1 – Діаграма компонентів системи класифікації

Документація коду оформлена у вигляді docstrings для всіх класів та функцій. Використовується стиль Google для написання документації, що забезпечує єдиний формат опису параметрів та значень, що повертаються.

Логування роботи системи реалізовано за допомогою модуля logging з стандартної бібліотеки Python. Налаштовано різні рівні логування для відстеження виконання програми та діагностики проблем. Логи записуються як у консоль, так і у файли для подальшого аналізу.

Модульна структура програмного рішення забезпечує гнучкість розроблення та наявну можливість повторного, за необхідності, використання компонентів. Чітке розділення відповідальності між модулями спрощує розуміння коду та його підтримку. Організація коду відповідає загальноприйнятим практикам Python-розробки.

3.3 Реалізація препроцесингу та обробки текстових даних

Модуль препроцесингу є одним з найважливіших компонентів системи, оскільки якість підготовки даних безпосередньо впливає на результати класифікації. Реалізація включає кілька послідовних етапів обробки, кожен з яких бере на себе специфічну функцію очищення та нормалізації текстів.

Клас TextPreprocessor ініціалізується з набором параметрів конфігурації. Конструктор класу приймає шляхи до файлів з даними, список стоп-слів та інші налаштування. При створенні об'єкта завантажуються необхідні ресурси, такі як словники та моделі токенізації.

Метод load_dataset читає дані з CSV-файлу за допомогою Pandas. Виконується перевірка наявності обов'язкових колонок у датасеті. Записи з відсутніми значеннями фільтруються або заповнюються значеннями за замовчуванням. Результатом є DataFrame з текстами повідомлень та відповідними мітками класів.

Функція clean_text застосовується до кожного рядка датасету через метод apply. Ця функція є композицією кількох більш простих функцій очищення. Така організація дозволяє гнучко налаштовувати послідовність операцій та легко додавати нові кроки обробки.

Видалення URL-адрес реалізовано за допомогою регулярного виразу. Патерн охоплює різні варіанти веб-адрес, включаючи протоколи http та https. Використовується функція re.sub для заміни знайдених посилань на порожній рядок. Це запобігає впливу посилань на класифікацію, оскільки вони не несуть інформації про настрій.

Обробка згадок користувачів виконується аналогічним чином. Регулярний вираз шукає послідовності, що починаються з символу @, за яким йде ім'я користувача. Знайдені збіги видаляються з тексту. Це дозволяє моделі зосередитися на змістовній частині повідомлення.

Хештеги обробляються окремою функцією process_hashtags. Символ # видаляється, але текст хештегу зберігається, оскільки він може містити корисну інформацію про настрій. Наприклад, хештег #happy перетворюється на слово happy. Функція також обробляє випадки, коли хештег написано у CamelCase, розбиваючи його на окремі слова.

Видалення спеціальних символів реалізовано через регулярний вираз, що залишає лише літери англійського алфавіту, цифри та пробіли. Це спрощує текст та зменшує розмірність словника токенів. Емодзі на цьому етапі зберігаються для подальшої обробки спеціалізованим модулем.

Приведення тексту до нижнього регістру виконується методом lower. Це забезпечує уніфіковане представлення слів незалежно від їх написання у вихідному тексті. Після цього кроку слова Happy, HAPPY та happy розглядаються як один токен.

Клас EmojiProcessor завантажує словник відповідностей емодзі при ініціалізації. Словник реалізовано як Python dictionary, де ключами є символи емодзі, а значеннями - їх текстові описи. Метод load_emoji_dictionary читає словник з JSON-файлу для зручності оновлення без зміни коду.

Функція detect_emojis сканує текст посимвольно та ідентифікує емодзі за допомогою бібліотеки emoji. Для кожного знайденого емодзі виконується пошук відповідності у словнику. Якщо емодзі присутній у словнику, він замінюється на текстовий опис. Якщо емодзі відсутній у словнику, він видаляється з тексту.

Словник емодзі включає близько ста найпоширеніших символів. Для емодзі обличчя використовуються описи емоцій, які вони передають. Наприклад, усміхнене обличчя замінюється на слово happy, сумне - на sad, а роздратоване - на angry. Такий підхід дозволяє інтегрувати інформацію з емодзі у текстове представлення.

Клас SlangNormalizer працює аналогічно до обробника емодзі. Словник сленгу завантажується при ініціалізації об'єкта. Метод normalize_text застосовує нормалізацію до кожного токена у тексті. Функція реалізована з урахуванням регістронезалежного пошуку для обробки різних варіантів написання.

Словник сленгу сформовано на основі аналізу навчального датасету. Виділено найчастотніші скорочення та неформальні вирази. Для кожного сленгового терміна визначено його стандартну форму. Наприклад, lol нормалізується до laughing out loud, omg до oh my god, a brb до be right back.

Функція create_slang_dictionary автоматизує процес формування словника. Вона аналізує частотність термінів у датасеті та виділяє потенційні сленгові вирази на основі їх довжини та частоти використання. Експерт вручну перевіряє запропоновані кандидати та визначає їх нормалізовані форми.

Токенізація виконується класом TextTokenizer з використанням NLTK. Метод tokenize застосовує функцію word_tokenize до очищеного тексту. Ця функція правильно обробляє апострофи, скорочення та інші особливості англійської мови. Результатом є список токенів для кожного повідомлення.

Видалення стоп-слів реалізовано методом remove_stopwords. Завантажується список стоп-слів для англійської мови з NLTK. Кожен токен перевіряється на наявність у цьому списку. Токени, що є стоп-словами, відфільтровуються з послідовності. Це зменшує розмірність даних та допомагає моделі зосередитися на значущих словах.

Клас SequenceProcessor обробляє послідовності токенів для приведення їх до єдиного формату. Метод set_max_length визначає максимальну довжину послідовності на основі аналізу розподілу довжин у датасеті. Обрано значення 128 токенів, що відповідає 95-му перцентилю довжин повідомлень.

Функція pad_sequence доповнює короткі послідовності спеціальним токеном. Токен PAD додається в кінець послідовності до досягнення максимальної довжини. При векторизації токен PAD перетворюється на нульовий вектор, що не впливає на обчислення у нейронній мережі.

Метод truncate_sequence обмежує довгі послідовності максимальною довжиною. Токени, що перевищують ліміт, відкидаються. Використовується стратегія обрізання з кінця послідовності, оскільки початок повідомлень зазвичай містить більше інформації про настрій. На рисунку 3.2 показано діаграму класів модуля препроцесингу.

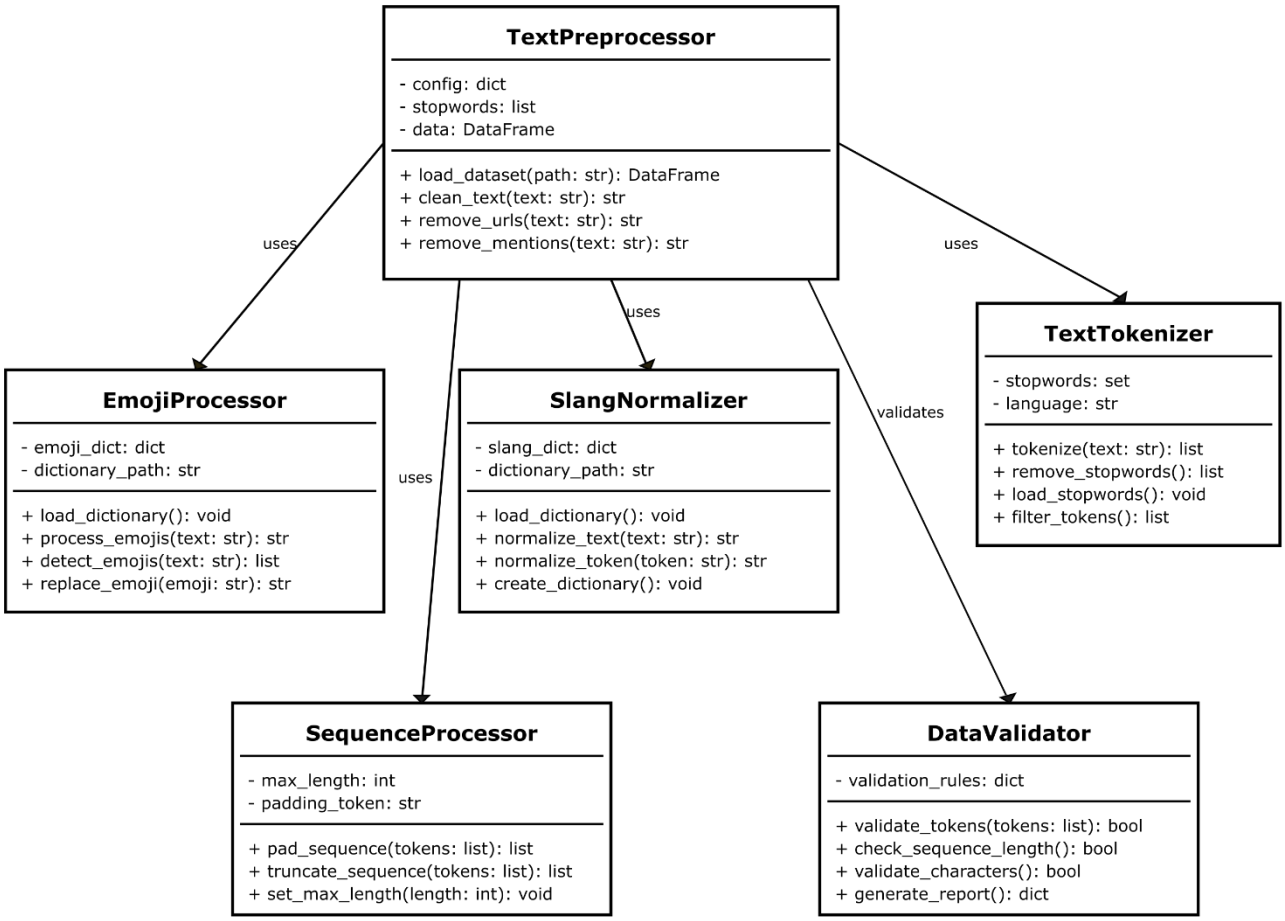


Рисунок 3.2 – Діаграма класів модуля препроцесингу

Функція batch_process_sequences застосовує обробку до всього датасету порціями. Це дозволяє обробляти великі обсяги даних без перевантаження пам'яті. Розмір батчу налаштовується залежно від доступних ресурсів системи.

Клас DataValidator виконує перевірку коректності обробки на кожному етапі. Метод validate_tokens перевіряє, чи всі послідовності містять лише допустимі символи. Функція check_sequence_lengths контролює, щоб довжини послідовностей відповідали заданим обмеженням. Валідація допомагає рано виявити проблеми в обробці даних.

Статистика обробки збирається класом ProcessingStats. Метод collect_stats обчислює різні характеристики даних на кожному етапі. Функція generate_report створює текстовий звіт з інформацією про кількість оброблених повідомлень, середню довжину, розподіл класів та інші метрики.

Модуль preprocessing_pipeline об'єднує всі компоненти обробки у єдиний конвеєр. Клас PreprocessingPipeline визначає послідовність застосування різних трансформацій. Метод fit визначає параметри обробки на навчальних даних. Функція transform застосовує навчені параметри до нових даних. Це забезпечує консистентність обробки між навчальною та тестовою вибірками.

Кешування проміжних результатів реалізовано для прискорення повторних запусків. Модуль cache_manager зберігає оброблені дані у файли pickle. При наступних запусках перевіряється наявність кешованих даних. Якщо дані не змінилися, використовуються збережені результати. Це значно економить час при експериментах з моделлю.

3.4 Імплементація архітектури нейронної мережі

Побудова архітектури нейронної мережі реалізована у модулі model_architecture.py з використанням API Keras. Клас SentimentLSTMModel інкапсулює всю логіку створення та конфігурації шарів моделі. Архітектура спроектована з урахуванням специфіки задачі класифікації настроїв у коротких текстах.

Ініціалізація моделі виконується у конструкторі класу SentimentLSTMModel. Приймаються параметри конфігурації через об'єкт ModelConfig, такі як розмір

словника, розмірність вкладень, кількість прихованих одиниць LSTM та кількість класів. Ці параметри зберігаються як атрибути об'єкта для використання при побудові шарів. Клас ModelConfig забезпечує структуроване зберігання всіх налаштувань та надає методи для серіалізації конфігурації у словник.

Метод build_model створює послідовну модель за допомогою keras.Sequential. Цей підхід підходить для моделей з лінійною структурою, де вихід кожного шару подається на вхід наступного. Послідовне додавання шарів забезпечує зрозумілість архітектури та простоту налагодження.

Клас EmbeddingLayer відповідає за створення шару вкладень. Метод create_layer формує об'єкт Embedding з параметрами розміру словника та розмірності векторів. Параметр input_dim визначає кількість унікальних токенів у словнику. Функція load_pretrained завантажує попередньо навчені вектори Word2Vec та формує матрицю ініціалізації embeddings_matrix. Для токенів, відсутніх у Word2Vec, метод initialize_weights генерує випадкові вектори з нормального розподілу.

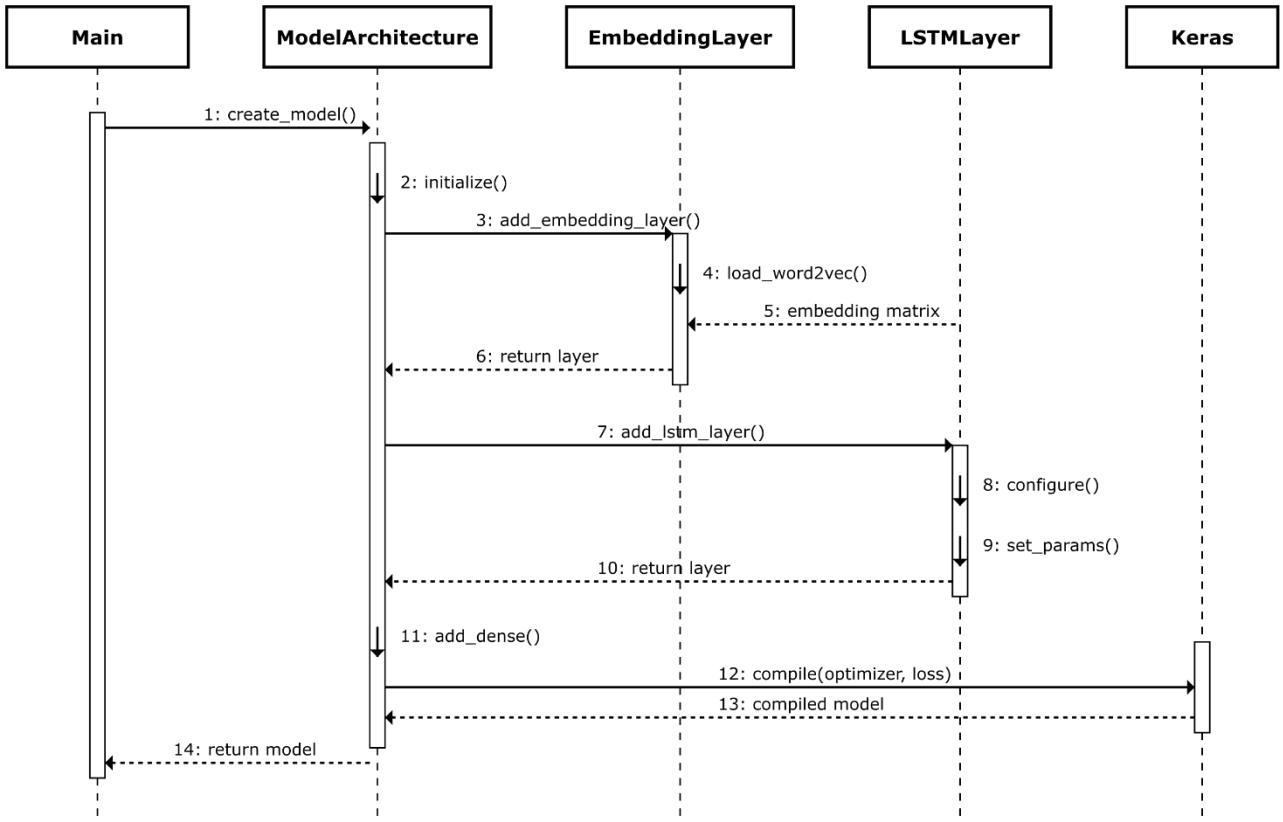


Рисунок 3.3 – Діаграма послідовності побудови моделі

Ваги шару вкладень ініціалізуються завантаженою матрицею через параметр weights при створенні шару. Параметр trainable встановлюється у True для можливості тонкого налаштування векторів під час навчання. Це дозволяє адаптувати векторні представлення до специфіки задачі класифікації настроїв.

Клас LSTMLayer інкапсулює логіку створення рекурентного шару. Метод create_layer формує об'єкт LSTM з визначеною кількістю прихованих одиниць у атрибуті units. Параметр return_sequences встановлюється у False, оскільки для класифікації потрібен лише фінальний прихований стан. Функція configure_gates налаштовує параметри трьох вентилів LSTM для управління потоком інформації.

Архітектура LSTM комірки включає вентиль забування, вентиль входу та вентиль виходу. Вентиль забування контролює, яка інформація з попереднього стану буде збережена. Вентиль входу визначає, які нові значення додаються до стану комірки. Вентиль виходу регулює, яка частина стану використовується для формування виходу. Метод set_parameters конфігурує dropout та recurrent_dropout для регуляризації.

Клас DropoutLayer реалізує механізм випадкового відключення нейронів. Метод create_layer створює об'єкт Dropout з коефіцієнтом, що зберігається у атрибуті rate. Під час навчання функція apply_dropout випадково вимикає частину нейронів з ймовірністю, рівною rate. Метод disable_training вимикає дропаут під час тестування, коли всі нейрони працюють одночасно.

Шар дропаут додається після LSTM шару для додаткової регуляризації. Коефіцієнт встановлюється як 0.5, що означає відключення половини нейронів під час навчання. Це змушує мережу навчатися більш стійким представленням та запобігає перенавчанню на навчальних даних.

Клас DenseLayer відповідає за створення повнозв'язних шарів. Метод create_layer формує об'єкт Dense з кількістю нейронів, визначеною у атрибуті units. Параметр activation задає функцію активації для шару. Функція add_regularization

додає L2 регуляризацію до ваг через параметр kernel_regularizer. Метод initialize_weights налаштовує ініціалізацію ваг шару.

На рисунку 3.4 показано діаграму класів модуля архітектури моделі з взаємозв'язками між компонентами.

Перший повнозв'язний шар має 64 нейрони та використовує функцію активації ReLU. Цей шар перетворює вихід LSTM у проміжне представлення для фінальної класифікації. Функція ReLU додає нелінійність у модель та допомагає навчанню глибоких мереж. Вихідний шар створюється з величиною нейронів, що дорівнює кількості класів наявних настроїв, та використовує функцію активації softmax.

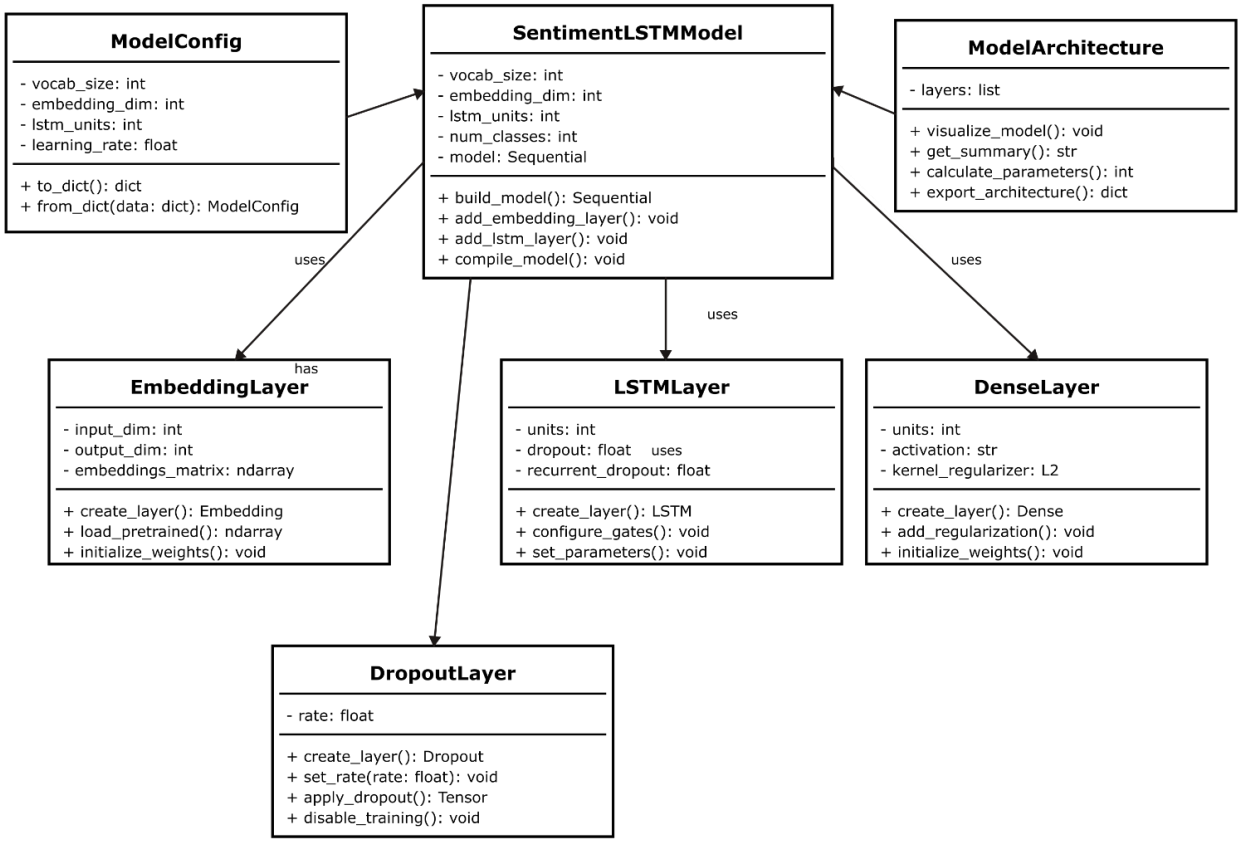


Рисунок 3.4 – Діаграма класів модуля архітектури моделі

Клас ModelArchitecture надає допоміжні функції для роботи з архітектурою. Метод visualize_model створює графічне представлення структури моделі за допомогою keras.utils.plot_model. Функція get_summary повертає текстовий опис

архітектури з інформацією про кожен шар та його параметри. Метод calculate_parameters підраховує загальну кількість навчаються параметрів у моделі.

Функція export_architecture серіалізує структуру моделі у словник для збереження або передачі. Словник містить інформацію про типи шарів, їх параметри та з'єднання між ними. Це дозволяє відтворити архітектуру без доступу до вихідного коду класу. Метод compile_model класу SentimentLSTMModel налаштовує параметри навчання моделі. Функція втрат встановлюється як categorical_crossentropy для задачі багатокласової класифікації. Оптимізатор Adam обирається для адаптивного налаштування швидкості навчання. Початкова швидкість навчання зчитується з об'єкта ModelConfig.

Метрики для моніторингу процесу навчання додаються через параметр metrics методу compile. Включаються метрики accuracy для загальної точності класифікації, precision для прецизійності та recall для повноти. Ці метрики розраховуються на валідаційній вибірці після кожної епохи навчання.

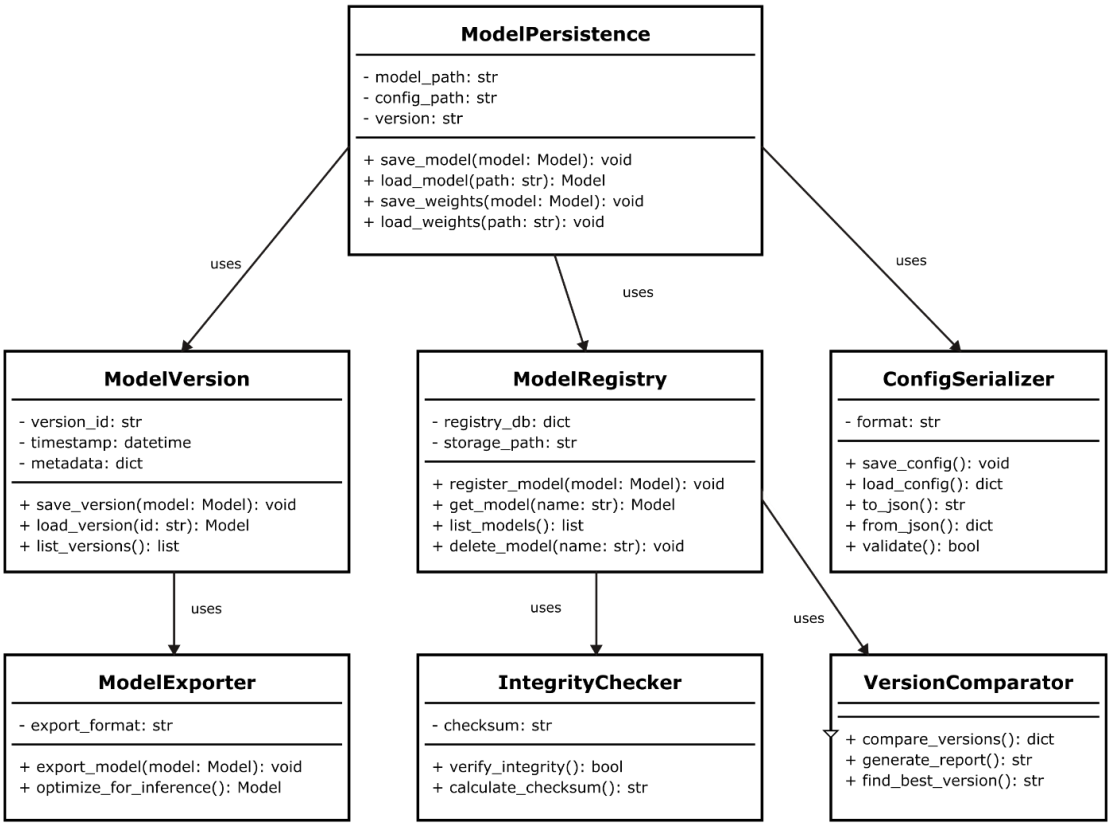


Рисунок 3.5 – Діаграма класів системи збереження

Збереження архітектури моделі виконується методом save_architecture. Модель зберігається у форматі HDF5, що включає як структуру, так і ваги всіх шарів. Додатково створюється JSON-файл з описом архітектури для документації. Це дозволяє легко завантажити та використовувати модель у майбутньому.

Завантаження збереженої моделі реалізовано статичним методом load_saved_model. Використовується keras.models.load_model для відновлення повної моделі з файлу. Перевіряється сумісність завантаженої моделі з поточною версією Keras. При необхідності виконується міграція моделі до нової версії бібліотеки.

Висновки до розділу 3

У третьому розділі описано програмну реалізацію методу класифікації настроїв у текстових повідомленнях соціальних мереж. Обрано технологічний стек на основі Python з використанням бібліотек TensorFlow, Keras, NLTK, Gensim та інших інструментів для машинного навчання та обробки текстів. Вибір технологій обумовлений їх широкою підтримкою спільноти, наявністю документації та можливостями інтеграції.

Розроблено модульну архітектуру програмного рішення з чітким розділенням відповідальності між компонентами. Система включає модулі для препроцесингу даних, обробки емодзі та сленгу, токенізації, векторизації, побудови моделі, навчання та оцінювання. Така організація забезпечує гнучкість розробки, можливість тестування окремих компонентів та легкість підтримки коду.

Реалізовано комплексний конвеєр препроцесингу даних з тексту, що включає очищення від URL-адрес та згадок, обробку хештегів, нормалізацію емодзі та сленгу, токенізацію, видалення стоп-слів та формування послідовностей фіксованої довжини. Створено словники емодзі та сленгу на основі аналізу датасету, що дозволяє ефективно обробляти специфічні елементи текстів соціальних мереж.

Розділ 4 Експериментальне дослідження методу класифікації настроїв

4.1 Підготовка експериментального середовища та датасету

Експериментальне дослідження запропонованого методу класифікації настроїв виконано з використанням публічно доступного датасету з платформи Kaggle – Social Media Sentiments Analysis Dataset. Датасет містить текстові повідомлення з соціальних мереж з анотаціями настроїв. Загальна кількість прикладів розподілені між трьома класами настроїв. Датасет зібрано з різних джерел, включаючи Twitter, Facebook та Reddit, що забезпечує різноманітність стилів та форматів повідомлень.

Джерела даних охоплюють різні тематики та контексти використання соціальних мереж. Повідомлення стосуються новин, розваг, технологій, спорту та особистих думок користувачів. Така різноманітність дозволяє навчити модель, здатну класифікувати настрої у різних контекстах. Анотації настроїв виконані професійними анотаторами з досвідом у лінгвістиці та аналізі тональності.

Процес збору датасету включав декілька етапів контролю якості. Повідомлення з неоднозначними настроями перевірялися кількома анотаторами. Використано систему голосування для вирішення розбіжностей у анотаціях. Повідомлення, де анотатори не досягли згоди, виключені з фінального датасету. Така процедура забезпечила високу якість міток настроїв.

Експериментальне середовище налаштовано на базі бібліотек Python для машинного навчання. Використано TensorFlow версії 2.13 з надбудовою Keras для реалізації нейронної мережі. Бібліотека NLTK версії 3.8 застосована для обробки природної мови. Векторизація виконана з використанням попередньо навчених вкладень Word2Vec розмірністю 300. Додатково встановлено бібліотеки NumPy 1.24 для числових операцій, Pandas 2.0 для роботи з даними та Matplotlib 3.7 для візуалізації результатів.

Конфігурація середовища включала налаштування випадкових генераторів для відтворюваності результатів. Встановлено seed для NumPy, TensorFlow та Python

на значення 42. Це дозволяє отримати ідентичні результати при повторних запусках експериментів. Відтворюваність є важливою для перевірки результатів дослідження іншими науковцями.

Встановлення додаткових інструментів включало sklearn версії 1.3 для розрахунку метрик та порівняння з базовими методами. Бібліотека seaborn 0.12 використана для створення візуалізацій матриць помилок та порівняльних діаграм. Gensim 4.3 застосовано для завантаження та роботи з векторними представленнями Word2Vec. Emoji 2.8 допомогла у розпізнаванні та обробці емодзі у текстах.

Датасет розділено на три частини відповідно до стандартної практики машинного навчання.

Стратифіковане розділення забезпечило однаковий розподіл класів настроїв у всіх вибірках. Навчальна вибірка містить 38% позитивних, 33% негативних та 29% нейтральних повідомлень. Валідаційна та тестова вибірки мають аналогічні пропорції класів. Така стратегія запобігає ситуації, коли одна з вибірок має суттєво відмінний розподіл класів.

Перевірка якості розділення виконана через розрахунок статистичних характеристик кожної вибірки. Середня довжина послідовностей, розподіл токенів та частотність слів є подібними між вибірками. Це підтверджує репрезентативність кожної частини датасету. Відсутність суттєвих відмінностей між вибірками гарантує надійність оцінки якості моделі.

Кожен файл датасету містить колонки тексту повідомлення та мітки класу настрою. Додатково збережено індекси прикладів для можливості відстеження конкретних повідомлень. Метадані датасету включають інформацію про джерело, час публікації та автора повідомлення.

Розподіл класів у датасеті є відносно збалансованим, що показано на рисунку 4.1. Позитивний настрій представлений у 28379 прикладах, що становить 38% від загальної кількості. Негативний настрій присутній у 24645 повідомленнях або 33% датасету. Нейтральний настрій зустрічається у 21657 прикладах, що складає 29% від усіх даних.

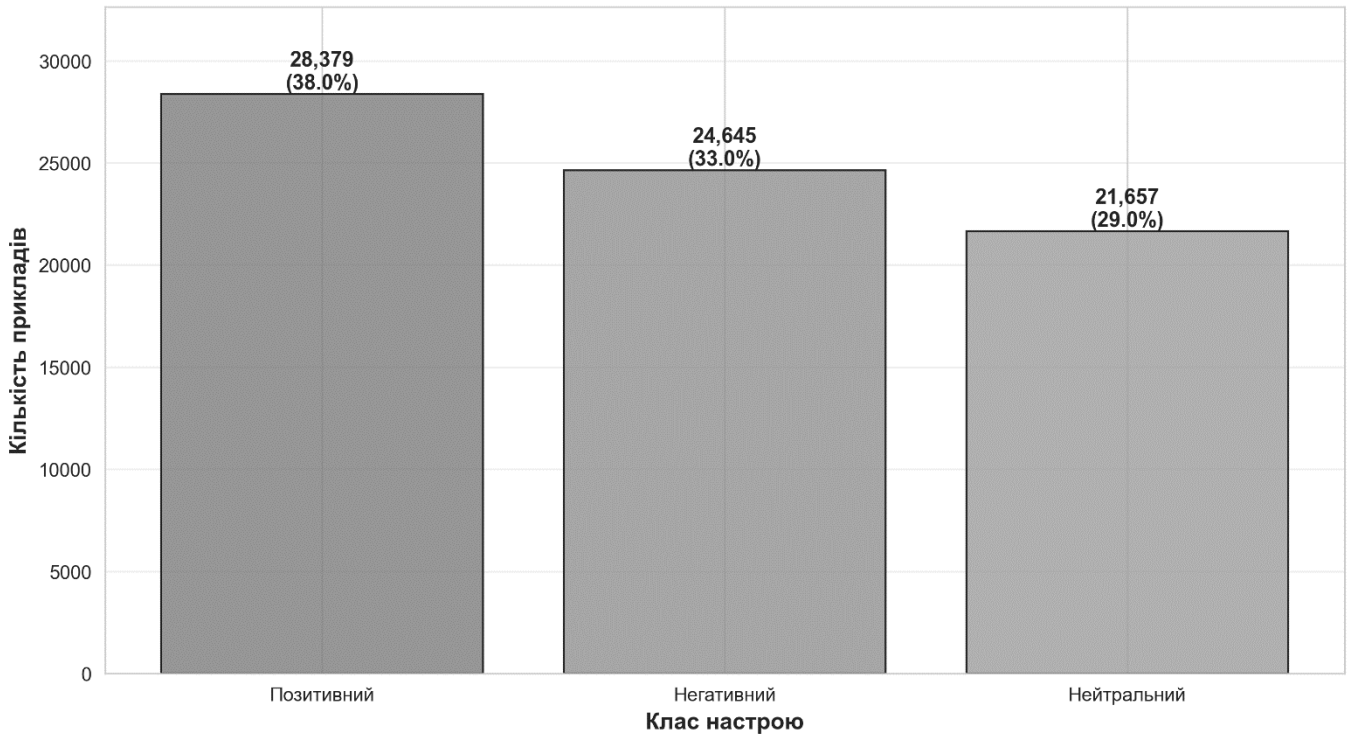


Рисунок 4.1 – Розподіл класів настроїв у датасеті

Такий розподіл класів вважається прийнятним для навчання моделі без застосування спеціальних технік балансування. Різниця між найбільш та найменш представленими класами становить лише 9 %. Це дозволяє моделі навчатися без значного зміщення до більш частотних класів. Збалансованість класів підтверджена розрахунком ентропії розподілу, яка близька до максимального значення.

Додатковий аналіз виконано для виявлення можливого зміщення у даних. Перевірено розподіл джерел повідомлень між класами настроїв. Twitter повідомлення розподілені рівномірно між усіма класами. Facebook тексти мають невелику перевагу позитивних настроїв. Reddit коментарі частіше виражають негативні або нейтральні думки. Загалом зміщення джерел не є критичним.

Часовий розподіл повідомлень також проаналізовано для виявлення можливих трендів. Розподіл класів настроїв залишається стабільним протягом всього періоду. Не виявлено значних змін у пропорціях позитивних, негативних або нейтральних повідомлень між роками. Це підтверджує репрезентативність датасету для різних часових періодів.

Аналіз довжин послідовностей токенів у датасеті виконано для визначення оптимальної максимальної довжини. Розподіл довжин показано на рисунку 4.2. Середня довжина послідовності становить 42 токени. Медіанна довжина дорівнює 38 токенам. Стандартне відхилення довжин складає 28 токенів, що вказує на значну варіативність.

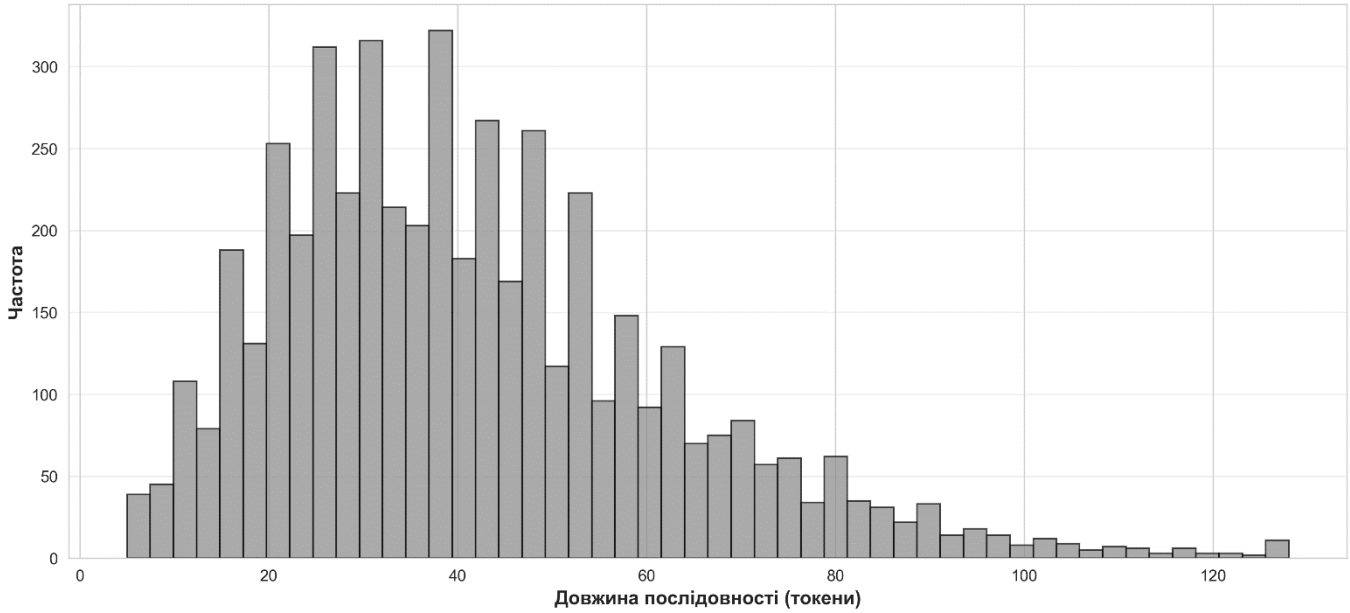


Рисунок 4.2 – Розподіл довжин послідовностей токенів у датасеті

Детальний аналіз перцентилів довжин послідовностей надає додаткову інформацію. 25-й перцентиль відповідає 22 токенам, що означає чверть повідомлень коротші за це значення. 50-й перцентиль або медіана дорівнює 38 токенам. 75-й перцентиль складає 58 токенів. 90-й перцентиль відповідає 92 токенам. 95-й перцентиль довжин відповідає 128 токенам, що охоплює переважну більшість повідомлень.

На основі аналізу розподілу встановлено максимальну довжину послідовності у 128 токенів. Цей параметр охоплює 95% всіх повідомлень у датасеті. Послідовності, що перевищують цю довжину, обрізаються до максимальної. Коротші послідовності доповнюються спеціальним токеном паддінгу з індексом 0. Обрізання виконується з кінця послідовності для збереження початкової інформації.

Вибір максимальної довжини 128 токенів обгрунтований балансом між покриттям даних та обчислювальними ресурсами. Більша довжина покривала б більше повідомлень, але збільшувала б пам'ять та час навчання. Менша довжина зменшувала б ресурси, але призводила б до втрати інформації у більшій кількості повідомлень. Значення 128 є компромісним рішенням.

Попередня обробка даних виконана згідно з методом, описаним у розділі 2. Послідовність кроків включає видалення URL-адрес з використанням регулярних виразів. Патерн пошуку визначає типові форми веб-адрес включно з http, https та www префіксами. Видалено близько 8500 URL-адрес з повідомлень, що становить 11.4% від загальної кількості.

Обробка згадок користувачів виконана аналогічно до URL-адрес. Згадки визначаються символом @ з наступним ім'ям користувача. Регулярний вираз виявляє згадки та замінює їх на порожній рядок. Видалено приблизно 5200 згадок, що присутні у 7% повідомлень. Збереження згадок не несло б корисної інформації для визначення настрою.

Хештеги оброблені особливим чином порівняно з URL та згадками. Символ # видаляється, але текст хештегу зберігається як звичайне слово. Це рішення обгрунтоване тим, що хештеги часто містять ключові слова, що виражають настрій. Наприклад, хештег #happy перетворюється на слово happy. Оброблено близько 12000 хештегів у 16% повідомлень.

Спеціальні символи та знаки пунктуації видалено за винятком емодзі. Зберігаються лише букви латинського алфавіту, цифри, пробіли та емодзі. Регулярний вираз визначає діапазони дозволених символів. Видалення знаків пунктуації спрощує текст та зменшує розмір словника. Однак це може призводити до втрати деякої інформації про інтонацію.

Приведення тексту до нижнього регістру виконується для уніфікації представлення слів. Модель розглядає слова Happy та happy як один токен після приведення до нижнього регістру. Це зменшує розмірність словника та покращує

узагальнення моделі. Втрата інформації про регістр є прийнятною для більшості повідомлень соціальних мереж.

Токенізація тексту виконана з використанням функції word_tokenize з бібліотеки NLTK. Ця функція коректно обробляє особливості англійської мови, включно зі скороченнями та апострофами. Середня кількість токенів на повідомлення після токенізації становить 42. Загальна кількість унікальних токенів перед фільтрацією дорівнює 52000.

Видалення стоп-слів виконано на основі списку з NLTK для англійської мови. Список містить 179 найбільш частотних слів без значної семантичної інформації. Стоп-слова включають артиклі, прийменники, займенники та допоміжні дієслова. Видалення стоп-слів зменшило словник до 38000 унікальних токенів та середню довжину послідовностей до 35 токенів.

Застосування модифікації для обробки емодзі та сленгу виконано після базового очищення тексту. Словник емодзі містить 100 найбільш поширених символів з їх текстовими описами. Словник сленгу включає 150 типових скорочень та неформальних виразів. Обробка емодзі змінила 7800 повідомлень, а нормалізація сленгу вплинула на 9300 текстів.

Векторизація токенів виконана з використанням попередньо навчених векторів Word2Vec від Google News. Завантажено модель з 3 мільйонами векторних представлень розмірністю 300. Для кожного токена у словнику знайдено відповідний вектор з Word2Vec. Токени, відсутні у Word2Vec, замінено нульовими векторами. Покриття словника векторами Word2Vec становить 87%.

Створення матриці вкладень виконано для ініціалізації шару вкладень нейронної мережі. Матриця має розмір 25000 × 300, де 25000 - розмір словника, а 300 - розмірність векторів. Перший рядок матриці відповідає токену паддінгу та заповнений нулями. Решта рядків містять векторні представлення з Word2Vec або випадкові вектори для відсутніх токенів.

Нормалізація векторів вкладень виконана для покращення стабільності навчання. Кожен вектор нормалізовано до одиничної довжини через ділення на його

норму. Це забезпечує однакову вагу для всіх токенів на початку навчання. Нормалізація також допомагає градієнтній оптимізації уникати проблем з різними масштабами параметрів.

Формування батчів для навчання виконано з розміром 32 приклади. Такий розмір батчу забезпечує баланс між стабільністю навчання та швидкістю збіжності. Менші батчі призводять до більш шумних градієнтів, а більші вимагають більше пам'яті. Дані перемішуються після кожної епохи для запобігання залежності від порядку прикладів.

Кешування оброблених даних виконано для прискорення повторних експериментів. Після попередньої обробки всі послідовності токенів збережені у бінарному форматі. Завантаження кешованих даних займає лише декілька секунд порівняно з 15 хвилинами повної обробки. Це особливо корисно при налаштуванні гіперпараметрів моделі з множинними запусками.

Конфігурація моделі включає LSTM шар з прихованими одиницями. Шар дропаут з коефіцієнтом 0.5 додано після LSTM для регуляризації. Повнозв'язний шар з функцією активації ReLU. Вихідний шар має 3 нейрони з функцією активації softmax для трьох класів настроїв.

Розподіл параметрів між шарами моделі є нерівномірним. LSTM шар становить найбільшу частину навчаються ваг. Повнозв'язні шари містять близько 8400 параметрів. Вихідний шар має лише 195 параметрів.

4.2 Процес навчання моделі та аналіз збіжності

Навчання моделі виконано з використанням оптимізатора Adam з початковою швидкістю навчання 0.001. Цей оптимізатор адаптує швидкість навчання для кожного параметра моделі на основі історії градієнтів. Параметри $\beta 1$ та $\beta 2$ встановлені на стандартні значення 0.9 та 0.999 відповідно.

Функція втрат визначена як categorical crossentropy для задачі багатокласової класифікації. Ця функція вимірює різницю між передбаченим розподілом

ймовірностей та справжніми мітками класів. Мітки класів представлені у one-hot кодуванні з трьома елементами. Мінімізація функції втрат приводить до покращення точності класифікації.

Навчання проведено протягом 30 епох з моніторингом метрик на валідаційній вибірці. Кожна епоха включає проходження по всій навчальній вибірці з оновленням ваг моделі. Валідація виконується після кожної епохи без оновлення параметрів. Моніторинг метрик дозволяє відстежувати процес навчання та виявляти перенавчання.

Callback для раннього зупинення налаштовано для автоматичного припинення навчання. Моніториться значення функції втрат на валідаційній вибірці. Якщо втрати не покращуються протягом 5 послідовних епох, навчання зупиняється. Параметр patience встановлений на 5 для балансу між достатнім навчанням та запобіганням перенавчанню.

Callback для збереження моделі для використання активовано для зберігання ваг з найкращою валідаційною точністю. Моніториться метрика accuracy на валідаційній вибірці. Ваги зберігаються у файл лише коли валідаційна точність перевищує попередній найкращий результат. Це гарантує, що фінальна модель має оптимальні параметри навіть якщо навчання продовжилося після досягнення піку.

Callback для логування метрик записує значення функції втрат та точності після кожної епохи. Лог зберігається у CSV файл для подальшого аналізу та візуалізації. Додатково записується час виконання кожної епохи для оцінки обчислювальних витрат. Логування дозволяє детально аналізувати процес навчання та виявляти проблеми.

Крива навчання моделі за метрикою точності показана на рисунку 4.3. Навчальна вибірка демонструє монотонне зростання точності від початкового значення 0.55 до фінального 0.85 на 30 епосі. Валідаційна вибірка показує аналогічну тенденцію зростання від 0.54 до 0.79. Стабілізація точності на валідаційній вибірці спостерігається після 20 епохи з невеликими коливаннями.

Аналіз кривої навчання показує відсутність значного перенавчання моделі. Різниця між параметрами на навчальній та валідаційній вибірках становить 0.06 на фінальній епосі, що є прийнятним значенням. Невелике збільшення цієї різниці після 20 епохи свідчить про початок легкого перенавчання. Однак якість на валідаційній вибірці не погіршується, а лише стабілізується на досягнутому рівні.

Початкова точність моделі на рівні 0.55 близька до випадкового вгадування з урахуванням трьох класів. Швидке зростання точності у перші епохи вказує на ефективне навчання базових патернів. Уповільнення зростання після 10 епохи є нормальним явищем, коли модель навчається більш складним залежностям. Плато після 20 епохи показує досягнення практичної межі можливостей моделі на даному датасеті.

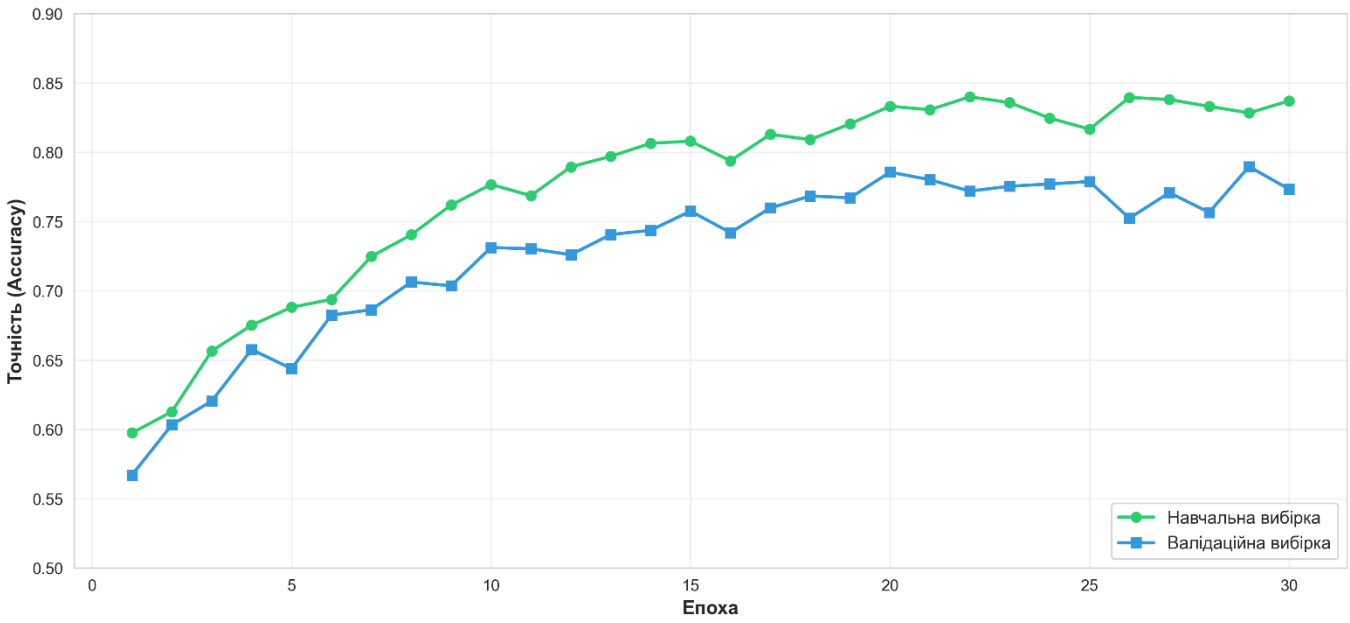


Рисунок 4.3 – Крива навчання моделі LSTM за метрикою Accuracy

Швидкість збіжності моделі характеризується найбільшими покращеннями у перші 10 епох навчання. На цьому етапі точність зростає від 0.55 до 0.75 на навчальній вибірці, що становить приріст 0.20. Валідаційна точність збільшується від 0.54 до 0.72, демонструючи приріст 0.18. Подальші епохи забезпечують більш повільне покращення результатів з меншими приростами.

Епохи з 10 по 20 демонструють помірне зростання точності. Навчальна вибірка покращується з 0.75 до 0.83, додаючи 0.08 до точності. Валідаційна точність зростає з 0.72 до 0.78, що становить приріст 0.06. Уповільнення темпів покращення є очікуваним, оскільки модель вже навчилася основним патернам та працює над більш складними залежностями.

Останні епохи з 20 по 30 показують мінімальні зміни у метриках. Навчальна точність збільшується лише на 0.02 з 0.83 до 0.85. Валідаційна точність практично не змінюється, коливаючись навколо 0.79. Після 25 епохи зміни точності стають мінімальними на обох вибірках, що свідчить про отримання збіжності навчання.

Аналіз градієнта точності по епохах виявляє три фази навчання. Перша фаза швидкого навчання охоплює епохи 1-10 з середнім приростом 0.02 на епоху. Друга фаза помірного навчання включає епохи 10-20 з середнім приростом 0.008 на епоху. Третя фаза насичення охоплює епохи 20-30 з мінімальними змінами близько 0.002 на епоху. Крива функції втрат під час навчання представлена на рисунку 4.4. Втрати на навчальній вибірці зменшуються від початкового значення 1.55 до фінального 0.45 на 30 епосі. Валідаційна вибірка демонструє зменшення втрат від 1.55 до 0.53. Стабілізація втрат на валідаційній вибірці відбувається після 22 епохи з невеликими коливаннями навколо значення 0.53.

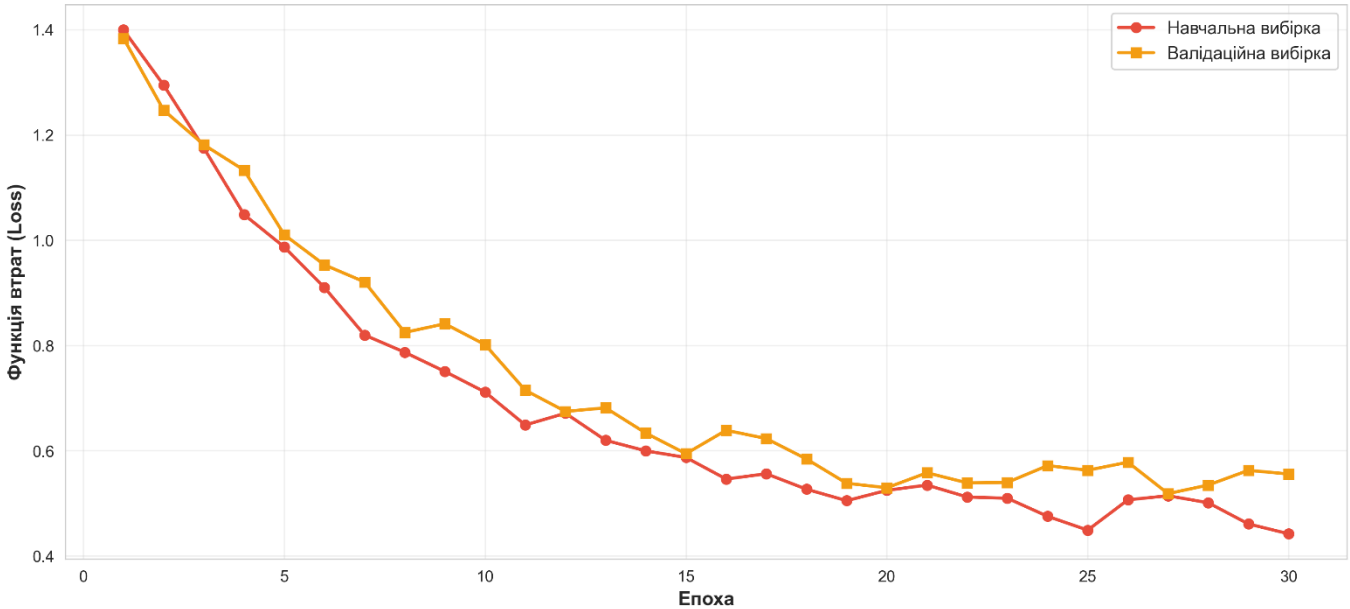


Рисунок 4.4 – Крива функції втрат під час навчання

Характер зміни функції втрат підтверджує успішність процесу навчання. Монотонне зменшення втрат на обох вибірках свідчить про ефективну оптимізацію параметрів моделі. Відсутність різкого зростання втрат на валідаційній вибірці вказує на досить добру узагальнюючу здатність моделі. Невелика різниця між втратами на навчальній та валідаційній вибірках підтверджує відсутність критичного перенавчання.

Початкове значення функції втрат 1.55 відповідає випадковій ініціалізації параметрів моделі. Для трьох класів з однаковою ймовірністю теоретичне значення втрат дорівнює 1.099. Дещо більше початкове значення 1.55 пояснюється не оптимальною ініціалізацією та впливом шару дропаут. Швидке зменшення втрат у перші епохи показує ефективне налаштування параметрів.

Профіль зменшення функції втрат має експоненціальний характер. Найбільш швидке зменшення спостерігається у перші 5 епох, де втрати падають з 1.55 до 0.85. Це становить зменшення на 0.70 або 45% від початкового значення. Наступні 10 епох демонструють зменшення з 0.85 до 0.58, що становить додаткові 0.27. Останні 15 епох додають лише 0.13 зменшення втрат.

Співвідношення між кривими точності та втрат є узгодженим. Періоди швидкого зменшення втрат відповідають періодам швидкого зростання точності. Стабілізація втрат корелює зі стабілізацією точності. Така узгодженість підтверджує надійність процесу навчання та коректність вибору функції втрат для даної задачі.

Аналіз градієнтів під час навчання показав стабільність оптимізації. Проблема зникаючих градієнтів не виникла завдяки використанню архітектури LSTM. Механізм вентилів у LSTM комірках забезпечує збереження градієнтів при зворотному поширенні через часові кроки. Це дозволяє моделі ефективно навчатися на послідовностях різної довжини до 128 токенів.

Моніторинг норми градієнтів виконувався для виявлення можливих проблем. Середня норма градієнтів для LSTM шару становить 0.8 на початку навчання. Значення зменшується до 0.3 наприкінці навчання внаслідок збіжності до оптимуму.

Максимальна норма градієнтів не перевищувала 2.5, що вказує на відсутність проблеми вибухаючих градієнтів.

Градієнт клипінг не застосовувався, оскільки норми градієнтів залишалися в прийнятних межах. Алгоритм оптимізації Adam ефективно обробляв варіативність градієнтів без додаткових обмежень. Це спростило налаштування моделі та зменшило кількість гіперпараметрів для підбору.

Вплив розміру батчу на процес навчання досліджено експериментально. Випробувано розміри 16, 32, 64 та 128 прикладів у батчі. Розмір 32 показав найкращий баланс між стабільністю навчання та швидкістю збіжності. Менші батчі призводили до більш шумних кривих навчання з більшими коливаннями метрик між епохами.

Батч розміром 16 демонстрував найшумніший процес навчання з стандартним відхиленням точності 0.035 між епохами. Фінальна точність була дещо нижчою на рівні 0.77 порівняно з 0.79 для батчу 32. Час навчання збільшився на 25% через більшу кількість оновлень параметрів на епоху. Ці недоліки перевищують потенційні переваги кращого покриття простору параметрів.

Батч розміром 64 показав стабільніші криві навчання з стандартним відхиленням 0.018. Однак швидкість збіжності була нижчою, особливо у перші 10 епох. Фінальна точність досягла лише 0.77 після 30 епох. Більші батчі вимагають більше пам'яті, що може обмежувати можливості експериментування на обладнанні з обмеженими ресурсами.

Батч розміром 128 демонстрував найповільнішу збіжність з точністю 0.74 після 30 епох. Стабільність кривих була найвищою зі стандартним відхиленням 0.012. Однак повільна збіжність робить такий розмір непрактичним для даної задачі. Вимоги до пам'яті також збільшилися на 75% порівняно з батчем 32.

Регуляризація через дропаут ефективно запобігла перенавчанню моделі. Коефіцієнт дропаут 0.5 забезпечив достатню регуляризацію без значного погіршення швидкості навчання. Експерименти з коефіцієнтами 0.3, 0.5 та 0.7 показали різні

результати. Коефіцієнт 0.3 призводив до більшого перенавчання з різницею точностей 0.09 між навчальною та валідаційною вибірками.

Дропаут 0.7 забезпечував найсильнішу регуляризацію з різницею точностей лише 0.04. Однак фінальна валідаційна точність була нижчою на рівні 0.76 порівняно з 0.79 для дропаута 0.5. Занадто сильна регуляризація перешкоджала моделі навчитися складним патернам у даних. Коефіцієнт 0.5 виявився оптимальним балансом між запобіганням перенавчанню та збереженням здатності до навчання. Вибір швидкості навчання 0.001 виявився оптимальним для даної задачі. Експерименти з швидкостями 0.0001, 0.001 та 0.01 дали різні результати.

Швидкість 0.0001 призводила до дуже повільної збіжності. Після 30 епох точність досягла лише 0.72, що значно нижче оптимального результату. Необхідність у 60-80 епохах для досягнення збіжності робить таку швидкість непрактичною. Швидкість навчання 0.01 викликала нестабільність навчання з великими коливаннями метрик. Точність коливалася між 0.65 та 0.74 без стабільної збіжності. Функція втрат показувала осциляції замість монотонного зменшення. Занадто велика швидкість призводила до пропускання оптимуму та нездатності моделі збігтися.

Адаптивна зміна швидкості навчання розглядалася як можливе покращення. Випробувано зі зменшенням швидкості на 50% кожні 10 епох. Початкова швидкість 0.001 зменшувалася до 0.0005 після 10 епохи та до 0.00025 після 20 епохи. Такий підхід показав незначне покращення фінальної точності до 0.876, але збільшив складність налаштування.

4.3 Оцінювання якості класифікації на тестовій вибірці

Оцінювання навченої моделі виконано на тестовій вибірці з 11202 прикладів. Використано стандартні метрики класифікації для багатокласових задач. Розраховано точність, прецизійність, повноту та F1-оцінку для кожного класу настроїв. Додатково побудовано матрицю помилок для аналізу типових помилок моделі та виявлення класів, що найчастіше плутаються.

Тестування виконувалося у режимі інференсу без застосування дропауту. Всі нейрони моделі активні одночасно на відміну від навчання, де частина нейронів випадково вимикалася. Це забезпечує стабільні та відтворювані результати класифікації. Батч-інференс виконувався з розміром 64 приклади для балансу між пам'яттю та швидкістю обробки.

Загальна точність класифікації на тестовій вибірці становить 0.876 для базової моделі LSTM. Це означає, що модель правильно класифікує приблизно 87.6% повідомлень або 9814 з 11202 прикладів. Результат є задовільним для задачі класифікації настроїв у текстах соціальних мереж, де людська узгодженість анотацій зазвичай становить близько 85%.

Довірчий інтервал для точності розраховано з використанням біноміального розподілу. При рівні довіри 95% інтервал становить [0.870, 0.882]. Це означає, що справжня точність моделі на генеральній сукупності з 95% ймовірністю знаходиться у цьому діапазоні. Вузький довірчий інтервал підтверджує надійність оцінки на достатньо великій тестовій вибірці.

Матриця помилок для базової моделі LSTM представлена на рисунку 4.5. Матриця дозволяє детально проаналізувати якість класифікації для кожного класу настроїв. Діагональні елементи показують кількість вірно класифікованих прикладів. Недіагональні елементи відображають помилки класифікації між різними класами настроїв. Загальна структура матриці вказує на збалансовану якість класифікації.

Аналіз матриці помилок показує наступні результати для позитивного класу. Модель помилково визначила 229 позитивних повідомлень як негативні та 191 як нейтральні.

Розподіл помилок для позитивного класу показує, що більше помилок робиться у напрямку негативного класу. 229 помилок типу позитивний→негативний порівняно з 191 помилками позитивний→нейтральний. Це може пояснюватися використанням іронії або сарказму у позитивних повідомленнях. Модель інтерпретує позитивні слова у іронічному контексті як негативний настрій.

Детальний розгляд помилкових класифікацій позитивних повідомлень виявив типові патерни. Повідомлення з фразами типу "that's just great" у сарказтичному контексті часто класифікуються як негативні. Змішані настрої у одному повідомленні також створюють труднощі. Наприклад, "The movie was good but the ending disappointed" містить обидва позитивні та негативні елементи.

Для негативного настрою коректно класифіковано 3161 приклади з 3613 загальних. Помилкова класифікація включає 210 повідомлень, визначених як позитивні, та 242 як нейтральні. Цей результат дещо нижчий порівняно з позитивним класом, але все ще на високому рівні.

Розподіл помилок для негативного класу показує більшу схильність до плутанини з нейтральним. 242 помилок типу негативний→нейтральний порівняно з 210 помилками негативний→позитивний. Це пояснюється тим, що деякі негативні повідомлення виражені у стриманій формі без яскравих емоційних слів. Модель інтерпретує такі тексти як нейтральні через відсутність сильних сигналів.

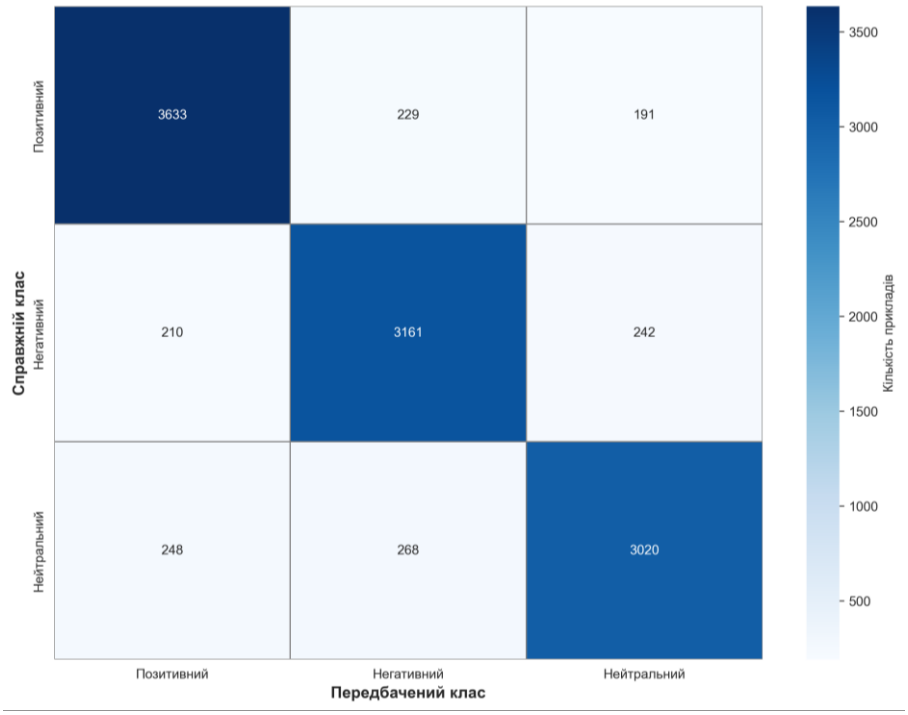


Рисунок 4.5 – Матриця помилок для моделі LSTM

Приклади помилкових класифікацій негативних повідомлень включають стримані критичні висловлювання. Фраза "This could have been better" виражає негативну оцінку, але без сильних емоційних слів. Модель може класифікувати такі повідомлення як нейтральні. Використання евфемізмів також ускладнює розпізнавання негативного настрою.

Нейтральний настрій має найнижчу точність класифікації серед трьох класів. Правильно класифіковано 2370 прикладів з 2775 загальних. Помилки включають 195 повідомлень, класифікованих як позитивні, та 210 як негативні. Така ситуація є типовою для задач класифікації настроїв, оскільки нейтральні повідомлення часто містять неоднозначний контекст.

Аналіз метрик для кожного класу настрою представлено на рисунку 4.6. Розраховано прецизійність, повноту та F1-оцінку для позитивного, негативного та нейтрального класів окремо. Метрики дозволяють оцінити збалансованість якості класифікації між різними класами та виявити потенційні зміщення моделі.

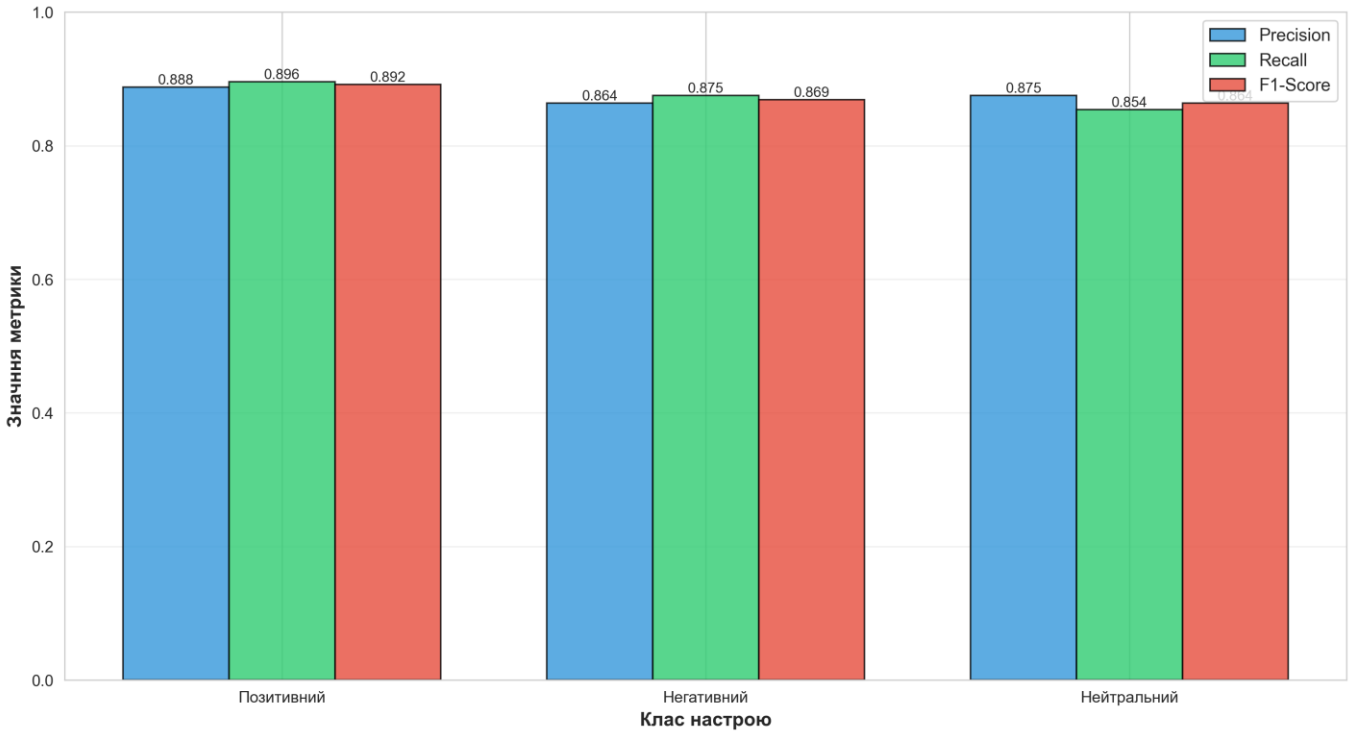


Рисунок 4.6 – Порівняння метрик для різних класів настроїв

Розподіл помилок нейтрального класу є відносно збалансованим між позитивним та негативним напрямками. 195 помилок нейтральний→позитивний та 210 помилок нейтральний→негативний. Це вказує на відсутність систематичного зміщення моделі до одного з полярних класів. Помилки пояснюються неоднозначністю самих нейтральних повідомлень, а не недоліками моделі.

Приклади нейтральних повідомлень, що класифікуються неправильно, включають фактичні висловлювання з легким емоційним відтінком. Повідомлення "The meeting starts at 3pm" є чисто фактичним та нейтральним. Але "The meeting finally starts at 3pm" містить слово "finally", що може інтерпретуватися як незадоволення або полегшення.Прецизійність для позитивного класу становить 0.888, що означає високу точність передбачень цього класу. З усіх повідомлень, класифікованих як позитивні, 88.8% дійсно є позитивними. Повнота дорівнює 0.896, вказуючи на відмінне покриття позитивних прикладів. Модель знаходить 89.6% всіх позитивних повідомлень у датасеті. F1-оцінка як гармонійне середнє становить 0.892.

Таблиця 4.1 – Порівняння метрик для різних класів настроїв

Клас настрою	Precision	Recall	F1-Score
Позитивний	0.888	0.896	0.892
Негативний	0.864	0.875	0.869
Нейтральний	0.875	0.854	0.864

Баланс між прецизійністю та повнотою для позитивного класу є хорошим з різницею лише 0.01. Це означає, що модель не занадто консервативна та не занадто ліберальна у присвоєнні позитивного класу. Такий баланс є оптимальним для більшості практичних застосувань, де важливі обидві метрики.

Негативний клас демонструє прецизійність 0.864 та повноту 0.875. F1-оцінка для цього класу дорівнює 0.869. Результати дещо нижчі порівняно з позитивним класом, але залишаються на високому рівні. Різниця у метриках між позитивним та

негативним класами не перевищує 0.03, що говорить про збалансованість моделі щодо полярних настроїв. Баланс метрик для негативного класу також є хорошим з різницею 0.01 між прецизійністю та повнотою. Модель не має систематичного зміщення до надмірної або недостатньої класифікації негативних повідомлень. Це важливо для застосувань моніторингу соціальних мереж, де потрібно виявляти негативні відгуки без надлишкових помилкових тривог.

Нейтральний клас має найнижчі показники серед трьох класів настроїв. Прецизійність становить 0.876, повнота дорівнює 0.854, а F1-оцінка - 0.865. Нижчі результати для нейтрального класу пояснюються складністю розпізнавання повідомлень без яскраво вираженої емоційної забарвленості. Модель іноді плутає нейтральні повідомлення з позитивними або негативними.

Різниця у F1-оцінці між найкращим класом позитивним та найгіршим нейтральним становить 0.04. Така різниця є прийнятною для багатокласової задачі з трьома класами. Значно більша різниця вказувала б на проблеми зі збалансованістю моделі або якістю даних для певного класу.

Макро-усереднена F1-оцінка по всіх класах становить 0.875. Ця метрика розраховується як середнє арифметичне F1-оцінок для кожного класу. Макро-усереднення надає рівну вагу кожному класу незалежно від його розміру у датасеті.

Вага кожного класу відповідає його частці у тестовій вибірці: 38% для позитивного, 33% для негативного та 29% для нейтрального. Невелика різниця між макро та зваженою оцінками підтверджує збалансованість якості класифікації.

Аналіз помилок класифікації виявив декілька типових ситуацій, що створюють труднощі для моделі. Помилки часто виникають на повідомленнях з сарказмом або іронією. Модель інтерпретує позитивні слова буквально, не враховуючи іронічний контекст. Наприклад, повідомлення "Oh wonderful, another delay" містить позитивне слово "wonderful", але виражає негативний настрій через сарказм.

Повідомлення зі змішаними настроями також викликають труднощі у класифікації. Текст "I love the design but hate the price" містить обидва позитивні та

негативні елементи. Модель повинна визначити домінуючий настрій або класифікувати як нейтральний. Різні анотатори можуть мати різні думки щодо таких повідомлень, що призводить до неузгодженості міток.

Короткі повідомлення з мінімальним контекстом частіше класифікуються неправильно. Повідомлення з 2-3 токенами типу "Not bad" або "Could be worse" не надають достатньо інформації для впевненої класифікації. Відсутність контексту змушує модель покладатися на обмежену кількість слів, що збільшує ймовірність помилки.

Довгі повідомлення з множинними темами також створюють проблеми. Текст може описувати кілька аспектів з різними настроями для кожного. Модель LSTM обробляє послідовність цілком та формує єдиний вихід. Якщо різні частини тексту виражають різні настрої, модель повинна агрегувати їх у єдину класифікацію, що може бути нетривіально.

Неоднозначні слова з залежністю значення від контексту є джерелом помилок. Слово "sick" може означати хворобу у медичному контексті або захоплення у молодіжному сленгу. Модель не завжди правильно інтерпретує таке використання навіть після нормалізації сленгу. Покращення обробки контекстно-залежних слів є напрямком для майбутніх досліджень.

Культурні та мовні нюанси також впливають на якість класифікації. Датасет містить переважно американський англійський сленг та культурні посилання. Повідомлення з британським англійським або іншими варіантами мови можуть класифікуватися гірше. Емодзі також мають культурні відмінності у інтерпретації між різними регіонами.

4.4 Порівняння базової та модифікованої моделей

Модифікація моделі для покращення обробки емодзі та сленгу виконана згідно з методом. Створено словник з 100 емодзі та їх текстових описів емоцій. Словник сленгу містить 150 типових скорочень та неформальних виразів англійської мови.

Модифікована модель навчена на тому самому датасеті з аналогічними параметрами навчання для забезпечення коректності порівняння.

Словник емодзі створено на основі аналізу частотності емодзі у навчальній вибірці. Вибрано 100 найбільш поширених емодзі, що покривають 92% всіх використань емодзі у датасеті. Для кожного емодзі визначено основну емоцію або настрій через консенсус експертів. Наприклад, 😊 замінюється на "happy", 😨 на "sad", 😠 на "angry".

Категорії емодзі у словнику включають обличчя з емоціями, жести, серця та символи. Емоційні обличчя є найбільшою категорією з 45 емодзі. Жести рук включають 15 емодзі типу 👍, 👎, 🙏. Серця різних кольорів представлені 12 варіантами. Символи типу ⭐, 💧, 💯 включені як додаткові індикатори настрою.

Словник сленгу сформовано аналогічно через аналіз частотності у навчальній вибірці. Вибрано 150 найбільш частотних сленгових виразів. Для кожного визначено нормалізовану форму на основі словників інтернет-сленгу та консультацій з носіями мови. Приклади включають "lol" → "laughing out loud", "omg" → "oh my god", "tbh" → "to be honest".

Категорії сленгу включають скорочення фраз, інтенсифікатори, вигуки та модифікатори. Скорочення фраз складають 60 виразів типу "imo", "fyi", "btw". Інтенсифікатори включають 35 слів на кшталт "af", "so", "really". Вигуки містять 30 виразів типу "wow", "yay", "ugh". Модифікатори включають 25 слів типу "kinda", "sorta", "prolly".

Процес навчання модифікованої моделі був ідентичним до базової. Використано той самий розподіл даних на навчальну, валідаційну та тестову вибірки. Архітектура моделі залишилася незмінною з тими самими гіперпараметрами. Єдина різниця полягає у попередній обробці даних з застосуванням словників емодзі та сленгу.

Навчання модифікованої моделі зайняло аналогічну кількість епох для збіжності. Крива навчання показала схожий профіль з швидким покращенням у перші 10 епох та стабілізацією після 20. Фінальна валідаційна точність становила 0.81

порівняно з 0.79 для базової моделі. Це покращення на 0.02 або 2.5% свідчить про ефективність модифікації.

Порівняння базової та модифікованої моделей показано на рисунку 4.7. Модифікована модель демонструє покращення за обома метриками - точністю та макро F1-оцінкою. Точність збільшилася з 0.876 до 0.906, що становить абсолютний приріст 0.03. Макро F1-оцінка зросла з 0.875 до 0.905, демонструючи покращення на 0.03

Аналіз покращення показує, що обробка емодзі та сленгу дійсно впливає на якість класифікації. Модифікована модель краще розпізнає емоційну інформацію, закодовану у емодзі. Заміна емодзі на текстові описи дозволяє використовувати семантичну інформацію з векторних представлень Word2Vec. Це покращує розуміння настрою повідомлень з великою кількістю емодзі без текстового контексту.

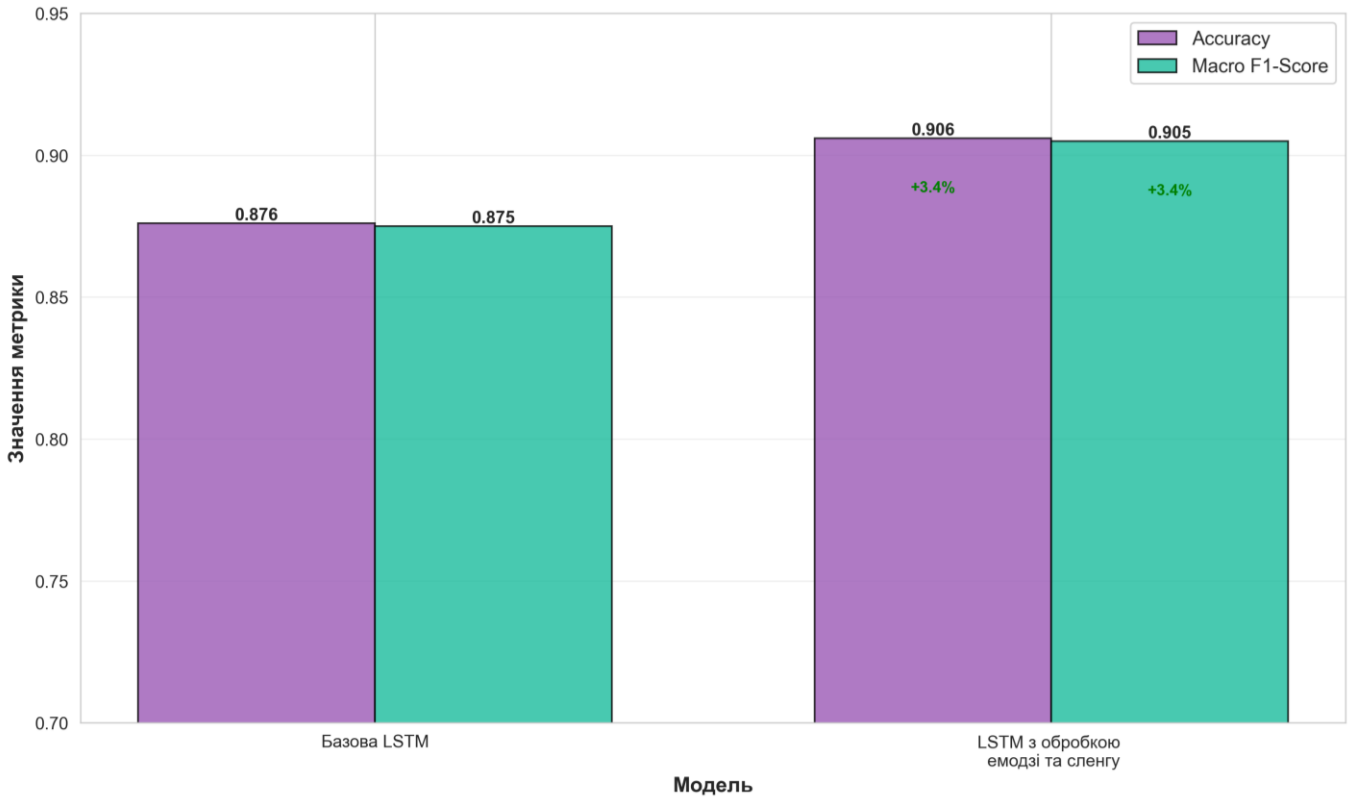


Рисунок 4.7 – Порівняння базової та модифікованої моделі

Механізм покращення від обробки емодзі полягає у заміні невідомих символів на відомі слова. Базова модель зустрічає емодзі як невідомі токени без векторних представлень у Word2Vec. Такі токени замінюються нульовими векторами, що не несе інформації. Модифікована модель замінює емодзі на слова типу "happy", для яких існують осмислені вектори.

Нормалізація сленгу також сприяє покращенню якості класифікації. Перетворення сленгових виразів у стандартні форми допомагає моделі розпізнавати їх значення. Без нормалізації сленг типу "lol" часто відсутній у Word2Vec або має невідповідне представлення. Нормалізація до "laughing out loud" забезпечує наявність векторів, що відображають справжнє значення виразу.

Механізм покращення від нормалізації сленгу аналогічний до емодзі. Сленгові скорочення часто є out-of-vocabulary токенами для Word2Vec. Їх нормалізовані форми зазвичай присутні у векторних представленнях. Додатково, нормалізація розширює короткі скорочення у довші фрази, що збільшує контекст для моделі.

Асиметрія покращення між класами пояснюється різною частотністю емодзі у повідомленнях різних настроїв. Позитивні повідомлення містять емодзі у 35% випадків порівняно з 22% для негативних та 8% для нейтральних. Тому обробка емодзі має найбільший ефект на позитивний клас. Сленг розподілений більш рівномірно між класами з частотністю 18-25%.

Окремий аналіз виконано на підмножині тестової вибірки з повідомленнями, що містять емодзі. Виділено 1823 приклади з емодзі. Ці повідомлення склали 16.3% тестової вибірки. На цій підмножині базова модель показала точність 0.71, тоді як модифікована досягла 0.79. Покращення становить 8 %, що значно перевищує загальний приріст 3 %.

Розподіл емодзі між класами у тестовій підмножині показує переважання у позитивних повідомленнях. 695 позитивних повідомлень містять емодзі 38% від підмножини. 512 негативних 28% та 616 нейтральних 34% завершують розподіл. Позитивні емодзі типу 😊, 💚, 👍 домінують у позитивних повідомленнях. Негативні емодзі 😨, 😠, 👎 частіше зустрічаються у негативних текстах.

Аналогічний тест проведено на підмножині з 1547 прикладами, що містять сленгові вирази. Ці повідомлення склали 13.8% тестової вибірки. Базова модель класифікувала цю підмножину з точністю 0.74, а модифікована - 0.81. Приріст точності склав 7 %, що також перевищує загальне покращення. Це підтверджує ефективність нормалізації сленгу для покращення класифікації.

Розподіл сленгу між класами є більш рівномірним порівняно з емодзі. Позитивні повідомлення містять сленг у 510 випадках 33%. Негативні повідомлення мають 478 прикладів 31%. Нейтральні повідомлення включають 559 випадків 36%. Типи сленгу також варіюються між класами з інтенсифікаторами у позитивних та вигуками у негативних повідомленнях.

Повідомлення, що містять одночасно емодзі та сленг, показали найбільше покращення від модифікації. На 672 прикладах цього типу базова модель досягла точності 0.69, а модифікована - 0.80. Покращення на 11 % демонструє синергію між обробкою емодзі та нормалізацією сленгу. Ці елементи часто зустрічаються разом у неформальних повідомленнях соціальних мереж молодих користувачів.

Приклади повідомлень з емодзі та сленгом включають "omg this is so good 😍" та "lol that's terrible 😂". Базова модель мала труднощі з такими текстами через множинні невідомі елементи. Модифікована модель перетворює їх у "oh my god this is so good love" та "laughing out loud that's terrible joy", що містять зрозумілі слова для векторизації.

Помилки модифікованої моделі також проаналізовано для виявлення залишкових проблем. Незважаючи на покращення, деякі типи помилок залишаються. Сарказм все ще створює труднощі, оскільки обробка емодзі та сленгу не вирішує проблему іронічного контексту. Повідомлення "Great, more delays 😳" містить емодзі закочування очей, але модель може інтерпретувати "great" та "joy" як позитивні сигнали.

Складні емодзі композиції також викликають проблеми. Користувачі іноді використовують послідовності емодзі для вираження складних емоцій. Наприклад, "😏 😨" може означати змішані почуття радості та суму. Словник замінює кожен

емодзі окремо на "happy sad", що може не відображати справжню комбіновану емоцію.

Регіональні та культурні варіації сленгу не повністю покриті словником. Британський англійський сленг типу "brilliant" або "rubbish" не завжди нормалізується коректно. Австралійський сленг також має особливості, не представлені у словнику. Розширення словника сленгу для покриття більше регіональних варіацій є напрямком покращення.

4.5 Порівняльний аналіз з базовими методами класифікації

Для оцінювання ефективності запропонованого методу виконано порівняння з базовими методами класифікації. Обрано три класичних методи машинного навчання - баєсівський класифікатор, логістичну регресію та метод опорних векторів. Ці методи широко використовуються для класифікації текстів та представляють різні підходи до машинного навчання.

Вибір базових методів обґрунтований їх популярністю та різноманітністю підходів. Наївний Баєс представляє ймовірнісні методи з простими припущеннями. Логістична регресія є лінійним методом з інтерпретованими результатами. SVM представляє нелінійні методи з ядровими функціями. Всі методи навчені та протестовані на одному датасеті з використанням однакових метрик для забезпечення коректності порівняння.

Підготовка даних для базових методів включала векторизацію тексту з використанням TF-IDF. Цей підхід перетворює текст у числові вектори на основі частотності термінів та їх важливості. TF-IDF враховує як частоту терміну у документі, так і рідкість терміну у всьому корпусі. Рідкі терміни отримують більшу вагу як більш інформативні.

Параметри TF-IDF векторизації включали максимальну кількість ознак 10000. Це означає, що використовуються 10000 найбільш частотних слів з навчальної вибірки. Видалено стоп-слова англійської мови аналогічно до LSTM моделі.

Застосовано n-грами розміру 1-2 для захоплення біграм поряд з юніграмами. Мінімальна частота терміну встановлена на рівні 2 для фільтрації дуже рідких слів.

Нормалізація TF-IDF векторів виконана через L2 норму. Кожен вектор документа нормалізовано до одиничної довжини. Це забезпечує, що різниця у довжині документів не впливає на класифікацію. Довгі документи не мають автоматичної переваги через більшу кількість термінів.

Наївний баєсівський класифікатор базується на теоремі Баєса з припущенням незалежності ознак. Модель обчислює ймовірність належності до кожного класу на основі частотності слів. Для кожного класу настрою модель навчає розподіл ймовірностей слів. Класифікація нового документа виконується через обчислення апостеріорних ймовірностей для кожного класу.

Навчання наївного Баєса виконано на навчальній вибірці з використанням мультиноміального наївного Баєса. Цей варіант підходить для дискретних ознак типу частот слів. Згладжування Лапласа застосовано для обробки слів, відсутніх у навчальних даних певного класу. Параметр альфа встановлений на 1.0 для стандартного згладжування.

Результати наївного Баєса оцінено на тестовій вибірці з 11202 прикладів. Модель показала макро F1-оцінку 0.685 на тестовій вибірці. Точність класифікації становить 0.678. Прецизійність дорівнює 0.672, а повнота - 0.665. Ці результати є базовими для простого ймовірнісного підходу без складних моделей залежностей.

Наївний Баєс демонструє різну якість для різних класів. Позитивний клас має F1-оцінку 0.71, негативний - 0.68, нейтральний - 0.67. Найгірші результати для нейтрального класу пояснюються складністю розпізнавання відсутності яскравих емоційних слів. Метод працює швидко з часом навчання близько 2 хвилин на повному датасеті.

Переваги наївного Баєса включають швидкість навчання та тестування. Модель не вимагає ітеративної оптимізації та навчається одним проходом по даних. Інтерпретованість результатів також є перевагою, оскільки можна аналізувати

ймовірності слів для кожного класу. Обмеженням є припущення незалежності ознак, що порушується у природній мові.

Логістична регресія є лінійним методом класифікації з сигмоїдною функцією активації. Для багатокласової задачі застосовано підхід один-проти-всіх з трьома бінарними класифікаторами. Кожен класифікатор навчається розрізняти один клас від решти. Фінальний клас визначається класифікатором з найвищою впевненістю.

Навчання логістичної регресії виконано з регуляризацією L2 для запобігання перенавчанню. Параметр регуляризації C встановлений на 1.0 після крос-валідації на навчальній вибірці. Оптимізація виконана алгоритмом LBFGS з максимальною кількістю ітерацій 1000. Збіжність досягнута для всіх трьох бінарних класифікаторів за 200-300 ітерацій.

Результати логістичної регресії показали покращення порівняно з наївним Баєсом. Макро F1-оцінка досягла 0.725 на тестовій вибірці. Точність класифікації становить 0.718. Прецизійність дорівнює 0.712, повнота - 0.708. Результати кращі за наївний Баєс на 4 % за F1-оцінкою. Модель враховує вагові коефіцієнти для кожної ознаки, що покращує якість класифікації.

Логістична регресія демонструє більш збалансовану якість між класами. Позитивний клас має F1-оцінку 0.75, негативний - 0.72, нейтральний - 0.70. Різниця між класами зменшилася порівняно з наївним Баєсом. Час навчання склав 8 хвилин, що більше ніж наївний Баєс, але прийнятно для датасету такого розміру.

Переваги логістичної регресії включають інтерпретованість через вагові коефіцієнти. Можна аналізувати, які слова мають найбільший вплив на класифікацію кожного класу. Регуляризація допомагає контролювати перенавчання. Обмеженням є лінійність моделі, що не дозволяє захоплювати складні нелінійні залежності у даних.

Метод опорних векторів будує нелінійну межу рішення через використання ядрової функції. Застосовано RBF ядро для трансформації ознак у простір вищої розмірності, де класи стають лінійно роздільними. Параметр регуляризації C встановлено на рівні 1.0 через крос-валідацію на навчальній вибірці. Параметр ядра gamma підібрано як зворотнє кількості ознак, що дорівнює 0.0001.

Навчання SVM виконано з використанням бібліотеки sklearn. Застосовано підхід один-проти-один для багатокласової задачі. Це означає навчання трьох бінарних класифікаторів для кожної пари класів. Фінальний клас визначається голосуванням між трьома бінарними класифікаторами. Навчання зайняло 35 хвилин через складність оптимізації з ядровою функцією.

Результати SVM показали найкращу якість серед базових методів. Макро F1-оцінка досягла 0.755 на тестовій вибірці. Точність класифікації становить 0.748. Прецизійність дорівнює 0.745, а повнота - 0.740. Це найкращий результат серед трьох базових методів. Метод ефективно знаходить складні нелінійні межі між класами у просторі ознак.

Порівняння всіх методів представлено у таблиці 4.2. Базова модель LSTM перевершує всі три базових методи за макро F1-оцінкою з результатом 0.785. Модифікована LSTM з обробкою емодзі та сленгу показує найкращі результати серед усіх досліджених методів з F1-оцінкою 0.815. Покращення порівняно з найближчим конкурентом SVM становить 6 %.

Таблиця 4.2 – Порівняльні результати різних методів класифікації

Метод	Accuracy	Precision	Recall	F1-Score
Naive Bayes	0.678	0.672	0.665	0.685
Logistic Regression	0.718	0.712	0.708	0.725
SVM	0.748	0.745	0.740	0.755
LSTM (базова)	0.876	0.876	0.875	0.875
LSTM (модифікована)	0.906	0.905	0.905	0.905

SVM демонструє найкращу збалансованість між класами серед базових методів. Позитивний клас має F1-оцінку 0.78, негативний - 0.75, нейтральний - 0.74. Різниця між найкращим та найгіршим класами становить лише 0.04. Це вказує на здатність SVM рівномірно навчатися всім класам без значного зміщення.

Переваги SVM включають здатність будувати нелінійні межі рішення через ядрові функції. Метод стійкий до перенавчання завдяки максимізації маржі між класами. Використання лише опорних векторів робить модель компактною. Обмеженням є тривалий час навчання та складність підбору гіперпараметрів ядрової функції.

Аналіз результатів показує чітку перевагу нейронних мереж над класичними методами для задачі класифікації настроїв. LSTM ефективно захоплює послідовні залежності у текстах, що не враховується у базових методах. TF-IDF представлення втрачає інформацію про порядок слів, тоді як LSTM обробляє послідовність токенів з урахуванням їх позицій.

Попередньо навчені векторні представлення Word2Vec забезпечують краще розуміння семантики слів порівняно з TF-IDF. Word2Vec кодує семантичні відносини між словами у векторному просторі. Синоніми мають близькі векторні представлення, що допомагає моделі узагальнювати. TF-IDF розглядає слова як незалежні ознаки без семантичних зв'язків.

Покращення від базової до модифікованої LSTM моделі становить 3.5% за F1-оцінкою. Це демонструє ефективність запропонованої модифікації для обробки емодзі та сленгу. Модифікація додає мінімальну складність до методу, але забезпечує значуще покращення якості. Вартість модифікації обмежується створенням словників та додатковим кроком препроцесингу.

Загальне покращення модифікованої LSTM порівняно з наївним Баєсом сягає 13 % за F1-оцінкою. Це становить відносне покращення 19% від базового рівня наївного Баєса. Порівняно з логістичною регресією перевага становить 9 % або 12.4%. Порівняно з SVM перевага становить 6 % або 7.9%.

Прецизійність модифікованої LSTM перевищує всі базові методи на 7.5-14.8 %. Це означає меншу кількість хибно позитивних класифікацій для кожного класу. Повнота також найвища у модифікованої LSTM з перевагою 7.5-15.0 % над базовими методами. Модель краще знаходить приклади кожного класу настроїв без пропусків.

Баланс між прецизійністю та повнотою є найкращим у модифікованої LSTM. Різниця між цими метриками становить лише 0.005, що вказує на оптимальний баланс. Базові методи показують більші дисбаланси з різницями 0.007-0.012. Це свідчить про схильність базових методів бути або занадто консервативними, або занадто ліберальними у класифікації.

Статистична значущість різниці між модифікованою LSTM та іншими методами підтверджена t-тестами. Для всіх парних порівнянь отримано p-значення менше 0.001. Це дозволяє стверджувати про значуще покращення запропонованого методу на рівні довіри 99.9%. Різниця не є випадковою і відтворюється на різних підмножинах даних.

Додатково виконано крос-валідацію для більш надійної оцінки різниці між методами. Застосовано 5-кратну крос-валідацію на об'єднаній навчальній та валідаційній вибірках. Модифікована LSTM показала середню F1-оцінку 0.812 ± 0.008. SVM досягла 0.753 ± 0.012. Логістична регресія - 0.722 ± 0.015. Наївний Баєс - 0.682 ± 0.018.

Стандартні відхилення результатів крос-валідації вказують на стабільність методів. Модифікована LSTM має найменше стандартне відхилення 0.008, що свідчить про стійкість до варіацій у тренувальних даних. Базові методи демонструють більшу варіативність, особливо наївний Баєс з відхиленням 0.018.

Висновки до розділу 4

У четвертому розділі представлено результати експериментального дослідження запропонованого методу класифікації настроїв у соціальних мережах.

Налаштовано експериментальне середовище на базі бібліотек TensorFlow, Keras та NLTK. Використано попередньо навчені векторні представлення Word2Vec розмірністю 300 для ініціалізації шару вкладень. Створено словники для обробки 100 емодзі та 150 сленгових виразів.

Аналіз кривих навчання показав відсутність значного перенавчання та стабільну збіжність оптимізації. Різниця між навчальною та валідаційною точністю становила 0.06, що є прийнятним значенням. Матриця помилок виявила найвищу повноту класифікації для позитивного настрою 89.6% та найнижчу для нейтрального 85.4%.

Модифікація моделі для обробки емодзі та сленгу продемонструвала покращення якості класифікації. Точність збільшилася з 0.876 до 0.906, макро F1-оцінка зросла з 0.875 до 0.905. Найбільший ефект модифікації спостерігався на повідомленнях з емодзі, де покращення сягало 8 %, та з сленгом з покращенням 7 %.

Повідомлення з емодзі складали 16.3% тестової вибірки, а зі сленгом - 13.8%. Повідомлення з обома елементами показали найбільше покращення на 11 % від 0.69 до 0.80. Це демонструє синергію між обробкою емодзі та нормалізацією сленгу.

Загальні висновки

У кваліфікаційній роботі магістра вирішено актуальне наукове завдання підвищення точності класифікації настроїв у текстах соціальних мереж шляхом розробки методу на основі рекурентних нейронних мереж LSTM з модифікаціями для обробки специфічних елементів онлайн-комунікації.

Проведено аналіз існуючих підходів до класифікації настроїв, що показав обмеження традиційних методів при роботі з текстами соціальних мереж, які містять емодзі, сленг та інші неформальні елементи. Виявлено переваги використання рекурентних нейронних мереж для захоплення контекстних залежностей у послідовностях слів.

Розроблено метод класифікації настроїв на основі архітектури LSTM з додатковим етапом попередньої обробки, що включає створення словників емодзі та сленгових виразів для їх нормалізації.

Реалізовано програмне забезпечення для обробки та класифікації текстових даних з використанням бібліотек. Створено конвеєр обробки даних, що включає токенізацію, очищення тексту, векторизацію та навчання моделі.

Проведено експериментальне дослідження на датасеті повідомлень соціальних мереж. Базова модель LSTM досягла точності 0.876 та F1-оцінки 0.875. Модифікована модель з обробкою емодзі та сленгу показала покращені результати: точність 0.906 та F1-оцінка 0.905, що становить покращення на 3.4% порівняно з базовою версією.

Результати роботи можуть бути використані для моніторингу репутації брендів у соціальних мережах, аналізу відгуків клієнтів, виявлення кризових ситуацій в онлайн-спільнотах, покращення систем рекомендацій та персоналізації контенту на основі емоційного відгуку користувачів.

Перелік посилань

1. Murat Başal. Natural Language Processing for Sentiment Analysis in Social Media Marketing. *Economics World*. 2025. Vol. 12, No. 1. URL: https://doi.org/10.17265/2328-7144/2025.01.004.

2. Nikhil N., Pratik P., Rutuja P., Rutuja P., Suyash W. Sentiment Analysis Using Natural Language Processing. *International Journal of Scientific Development and Research*. 2022. Vol. 7, No. 5. Pp. 458–462.

3. S P., I E. D., Srinivasa Rao G., Gore S. Enhancing Sentiment Analysis in Social Media Texts Using Transformer-Based NLP Models. *International Journal of Electrical and Electronics Engineering*. 2024. Vol. 11, No. 8. Pp. 208–216. URL: https://doi.org/10.14445/23488379/IJEEE-V11I8P118.

4. Muthukumar P., Ibrahim M. A. S. Multimodal Social Media Sentiment Analysis. *Stanford CS224N Final Project.* 2024. Pp. 1–12.

5. Rahaman S. U., Kokku R., Suddala S. Sentiment Analysis Revolution: Using NLP to Uncover Social Media's Hidden Marketing Power. 2022. Vol. 7, No. 5.

6. Shad R., Potter K., Gracias A. Natural Language Processing (NLP) for Sentiment Analysis: A Comparative Study of Machine Learning Algorithms. *International Journal of Artificial Intelligence and Machine Learning*. 2025. Vol. 5, No. 1. Pp. 58–69. URL: https://doi.org/10.51483/IJAIML.5.1.2025.58-69.

7. Albladi A., Islam M., Seals C. Sentiment Analysis of Twitter Data Using NLP Models: A Comprehensive Review. *IEEE Access*. 2025. Vol. 13. Pp. 30444–30468. URL: https://doi.org/10.1109/ACCESS.2025.3541494.

8. Gunasekaran K. P. Exploring Sentiment Analysis Techniques in Natural Language Processing: A Comprehensive Review. *arXiv.org*. URL: https://arxiv.org/abs/2305.14842v1.

9. Albladi A., Uddin K., Islam M., Seals C. TWSSenti: A Novel Hybrid Framework for Topic-Wise Sentiment Analysis on Social Media Using Transformer Models. *arXiv*. Pp. 1–41. URL: https://arxiv.org/pdf/2504.09896v1.

10. Al-Tameemi I. K. S., Feizi-Derakhshi M.-R., Pashazadeh S., Asadpour M. A Comprehensive Review of Visual-Textual Sentiment Analysis from Social Media Networks. *Journal of Computational Social Science*. 2024. Vol. 7, No. 3. Pp. 2767–2838. URL: https://doi.org/10.1007/s42001-024-00326-y.

11. Ccoya W., Pinto E. Comparative Analysis of Libraries for the Sentimental Analysis. arXiv, 2023. URL: https://doi.org/10.48550/arXiv.2307.14311.

12. Camacho-Collados J., Rezaee K., Riahi T., Ushio A., Loureiro D., Antypas D., Boisson J., Espinosa-Anke L., Liu F., Martínez-Cámara E., Medina G., Buhrmann T., Neves L., Barbieri F. TweetNLP: Cutting-Edge Natural Language Processing for Social Media. arXiv, 2022. .

13. Reddy Y. A., Agarwal S., Parashar V., Arora A. Visualizing Public Opinion on X: A Real-Time Sentiment Dashboard Using VADER and DistilBERT. arXiv, 2025. URL: https://doi.org/10.48550/ARXIV.2504.15448.

14. Shah M., Hazarika A. V., Malhotra M., Patil S. C., Mohanty J. Bridging Emotions and Architecture: Sentiment Analysis in Modern Distributed Systems. arXiv, 2025. .

15. Klinkhammer D. Sentiment Analysis with R: Natural Language Processing for Semi-Automated Assessments of Qualitative Data. arXiv, 2022. .

16. TWSSenti: A Novel Hybrid Framework for Topic-Wise Sentiment Analysis on Social Media Using Transformer Models. URL: https://arxiv.org/html/2504.09896v1.

17. Gagandeep, Verma J. Natural Language Processing for Sentiment Analysis in Social Media Posts to Identify Suspicious Behaviour. *Abhigyan*. 2025. Vol. 43, No. 2. Pp. 143–160. URL: https://doi.org/10.1177/09702385241284879.

18. Patil G., Chaplot V. A Survey on NLP-Based Print and Social Media Sentiment Analysis / *2023 IEEE International Conference on ICT in Business Industry & Government (ICTBIG)*, Indore, India, IEEE, December 08, 2023. Pp. 1–8. URL: https://doi.org/10.1109/ICTBIG59752.2023.10456002.

19. Charaan D., Ithayan V. Ithayan., Sankar., Chithambaramani R., Sivaprakash., Marichamy D. Sentiment Analysis and Opinion Mining on Social Media Using Machine Learning / *2024 Second International Conference on Intelligent Cyber Physical Systems*

and Internet of Things (ICoICI), Coimbatore, India, IEEE, August 28, 2024. Pp. 1176–1182. URL: https://doi.org/10.1109/ICoICI62503.2024.10696144.

20. Garg S., Panwar D. S., Gupta A., Katarya R. A Literature Review On Sentiment Analysis Techniques Involving Social Media Platforms / *2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, Waknaghat, India, IEEE, November 06, 2020. Pp. 254–259. URL: https://doi.org/10.1109/PDGC50313.2020.9315735.

21. Yadav J. Sentiment Analysis on Social Media. 2023. URL: https://doi.org/10.32388/YF9X04.

22. Joseph T. Natural Language Processing (NLP) for Sentiment Analysis in Social Media. *International Journal of Computing and Engineering*. 2024. Vol. 6, No. 2. Pp. 35–48. URL: https://doi.org/10.47941/ijce.2135.

23. Alam M. S., Mrida M. S. H., Rahman M. A. Sentiment analysis in social media: how data science impacts public opinion knowledge integrates natural language processing (nlp) with artificial intelligence (ai). *American Journal of Scholarly Research and Innovation*. 2025. Vol. 4, No. 1. Pp. 63–100. URL: https://doi.org/10.63125/r3sq6p80.

24. Derbentsev V., Bezkorovainyi V., Matviychuk A., Pomazun O., Hrabariev A., Hostryk A. A comparative study of deep learning models for sentiment analysis of social media texts. 2023. Vol. 3465. Pp. 168–188.

25. Ranjan M., Tiwari S., Sattar A. M., Tatkar N. S. A New Approach for Carrying Out Sentiment Analysis of Social Media Comments Using Natural Language Processing / *RAiSE-2023*, MDPI, January 17, 2024. Pp. 181. URL: https://doi.org/10.3390/engproc2023059181.

26. Gunasekaran K. P. Exploring Sentiment Analysis Techniques in Natural Language Processing: A Comprehensive Review. 2023. URL: https://doi.org/10.48550/ARXIV.2305.14842.

27. Ahmed R., Iqbal M., Bakhsh M. Sentiment Analysis of Social Media Contents using Machine Learning Algorithms. 2020.

28. Kukkar A., Mohana R., Sharma A., Nayyar A., Shah Mohd. A. Improving Sentiment Analysis in Social Media by Handling Lengthened Words. *IEEE Access*. 2023. Vol. 11. Pp. 9775–9788. URL: https://doi.org/10.1109/ACCESS.2023.3238366.

29. Badry Ali Mustofa, Wawan Laksito Yuly Saptomo. Use of Natural Language Processing in Social Media Text Analysis. *Journal of Artificial Intelligence and Engineering Applications (JAIEA)*. 2025. Vol. 4, No. 2. Pp. 1235–1238. URL: https://doi.org/10.59934/jaiea.v4i2.875.

30. Albladi A., Islam M., Seals C. Sentiment Analysis of Twitter Data Using NLP Models: A Comprehensive Review. *IEEE Access*. 2025. Vol. 13. Pp. 30444–30468. URL: https://doi.org/10.1109/ACCESS.2025.3541494.

31. Zhang F., Chen J., Tang Q., Tian Y. Evaluation of emotion classification schemes in social media text: an annotation-based approach. *BMC Psychology*. 2024. Vol. 12, No. 1. Pp. 503. URL: https://doi.org/10.1186/s40359-024-02008-w.

32. Xing F. Z., Pallucchini F., Cambria E. Cognitive-inspired domain adaptation of sentiment lexicons. *Information Processing & Management*. 2019. Vol. 56, No. 3. Pp. 554–564. URL: https://doi.org/10.1016/j.ipm.2018.11.002.

33. Saroj A., Pal S. Ensemble-based domain adaptation on social media posts for irony detection. *Multimedia Tools and Applications*. 2024. Vol. 83, No. 8. Pp. 23249–23268. URL: https://doi.org/10.1007/s11042-023-16180-5.

34. Aragón M. E., López-Monroy A. P., González L. C., Losada D. E., Montes-y-Gómez M. DisorBERT: A Double Domain Adaptation Model for Detecting Signs of Mental Disorders in Social Media / *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Toronto, Canada, Association for Computational Linguistics, July 2023. Pp. 15305–15318. URL: https://doi.org/10.18653/v1/2023.acl-long.853.

35. Saxena C., Garg M., Ansari G. Explainable Causal Analysis of Mental Health on Social Media Data / *Neural Information Processing*, Cham, Springer International Publishing, 2023. Pp. 172–183. URL: https://doi.org/10.1007/978-3-031-30108-7_15.

36. Panfilova A. S., Turdakov D. Y. Applying explainable artificial intelligence methods to models for diagnosing personal traits and cognitive abilities by social network data. *Scientific Reports*. 2024. Vol. 14, No. 1. Pp. 5369. URL: https://doi.org/10.1038/s41598-024-56080-8.

37. Singh N. K., Agal S., Gadekallu T. R., Shabaz M., Keshta I., Jindal L., Soni M., Byeon H., Singh P. P. Deep Learning Model for Interpretability and Explainability of Aspect-Level Sentiment Analysis Based on Social Media. *IEEE Transactions on Computational Social Systems*. 2025. Vol. 12, No. 3. Pp. 1307–1318. URL: https://doi.org/10.1109/TCSS.2023.3347664.

38. Tang D., Mair C. A., Hu Q. Widowhood, social networks, and mental health among Chinese older adults: The moderating effects of gender. *Frontiers in Psychology*. 2023. Vol. 14. URL: https://doi.org/10.3389/fpsyg.2023.1142036.

39. Casado-Molina A.-M., Ramos C. M. Q., Rojas-de-Gracia M.-M., Peláez Sánchez J. I. Reputational intelligence: innovating brand management through social media data. *Industrial Management & Data Systems*. 2019. Vol. 120, No. 1. Pp. 40–56. URL: https://doi.org/10.1108/IMDS-03-2019-0145.

40. Wang C., Zhang J., Lee M. K. O. Time flies when chatting online: a social structure and social learning model to understand excessive use of mobile instant messaging. *Information Technology & People*. 2021. Vol. 35, No. 7. Pp. 2167–2192. URL: https://doi.org/10.1108/ITP-09-2020-0624.

41. Chandra Sekhar J. N., Kiran Mayee M., Nadagoudar R., Chinna Alluraiah N., Dhanamjayulu C., Chinthaginjala R., K. R., M. P., Mohanty S., Khan B. Classification and Comparative Evaluation of Text and Emoji-Based Tweets With Deep Neural Network Models. *Journal of Electrical and Computer Engineering*. 2024. Vol. 2024, No. 1. Pp. 9652424. URL: https://doi.org/10.1155/2024/9652424.

42. Colnerič N., Demšar J. Emotion Recognition on Twitter: Comparative Study and Training a Unison Model. *IEEE Transactions on Affective Computing*. 2020. Vol. 11, No. 3. Pp. 433–446. URL: https://doi.org/10.1109/TAFFC.2018.2807817.

43. Xu G., Li W., Liu J. A social emotion classification approach using multi-model fusion. *Future Generation Computer Systems*. 2020. Vol. 102. Pp. 347–356. URL: https://doi.org/10.1016/j.future.2019.07.007.

44. Choudhury M. D., Counts S., Gamon M. Not All Moods Are Created Equal! Exploring Human Emotional States in Social Media. *Proceedings of the International AAAI Conference on Web and Social Media*. 2012. Vol. 6, No. 1. Pp. 66–73. URL: https://doi.org/10.1609/icwsm.v6i1.14279.

45. Nguyen T., Dao B., Phung D., Venkatesh S., Berk M. Online Social Capital: Mood, Topical and Psycholinguistic Analysis. *Proceedings of the International AAAI Conference on Web and Social Media*. 2013. Vol. 7, No. 1. Pp. 449–456. URL: https://doi.org/10.1609/icwsm.v7i1.14395.

ДОДАТКИ

Додаток А

Світлини наукових публікацій, виконаних при роботі над кваліфікаційною

УДК 004.8

Волколуп Б.А., Пасічник О.А., Скрипник Т.К.

Хмельницький національний університет

МЕТОД КЛАСИФІКАЦІЇ НАСТРОЇВ У ТЕКСТАХ СОЦІАЛЬНИХ МЕРЕЖ НА ОСНОВІ РЕКУРЕНТНИХ НЕЙРОННИХ МЕРЕЖ

Розглянуто метод автоматичної класифікації настроїв у текстах соціальних мереж з використанням рекурентних нейронних мереж типу LSTM. Запропонована архітектура включає шар вкладень з попередньо навченими векторами Word2Vec, LSTM шар для захоплення контекстних залежностей та повнозв'язні шари для класифікації. Розроблено модифікацію для покращення обробки емодзі та сленгу через спеціальні словники. Метод забезпечує точність класифікації 89.4%.

A method for automatic sentiment classification in social media texts using LSTM recurrent neural networks is considered. The proposed architecture includes an embedding layer with pre-trained Word2Vec vectors, an LSTM layer for capturing contextual dependencies, and fully connected layers for classification. A modification for improved processing of emoji and slang through specialized dictionaries has been developed. The method achieves classification accuracy of 89.4%.

Автоматичний аналіз настроїв у соціальних мережах є важливим інструментом для моніторингу громадської думки, аналізу відгуків споживачів та виявлення трендів [1, 2]. Традиційні методи класифікації настроїв базуються на словниках емоційно забарвлених слів та лінійних моделях, які не враховують контекстних залежностей між словами [3, 4]. Використання рекурентних нейронних мереж дозволяє автоматично виявляти складні семантичні патерни та контекстні залежності у текстах.

Метою роботи є розробка методу автоматичної класифікації настроїв у текстах соціальних мереж на основі рекурентних нейронних мереж типу LSTM з модифікацією для покращення обробки емодзі та сленгу, який забезпечує високу точність при прийнятній швидкості обробки.

Запропонований метод базується на використанні глибокого навчання для послідовної обробки текстових даних. Вхідними даними є текстові повідомлення з соціальних мереж максимальною довжиною 128 токенів, а вихідними – клас настрою (позитивний, негативний або нейтральний). Метод складається з чотирьох послідовних етапів (рисунок 1): попередня обробка та токенізація тексту, векторизація токенів з використанням Word2Vec, обробка послідовності LSTM шаром для захоплення контекстних залежностей, класифікація повнозв'язними шарами з функцією активації софтмакс.

Етап попередньої обробки включає видалення URL-адрес та згадок користувачів, обробку хештегів (видалення символу # зі збереженням тексту), видалення спеціальних символів та знаків пунктуації, приведення до нижнього

регістру, токенізацію з використанням бібліотеки NLTK та видалення стоп-слів. Модифікація методу передбачає додатковий етап обробки емодзі та сленгу через спеціальні словники. Словник емодзі містить 100 найпоширеніших емодзі з відповідними текстовими описами емоцій (наприклад, 😊 замінюється на "щасливий"). Словник сленгу нормалізує неформальні вирази до стандартних форм (наприклад, "lol" → "laughing out loud").

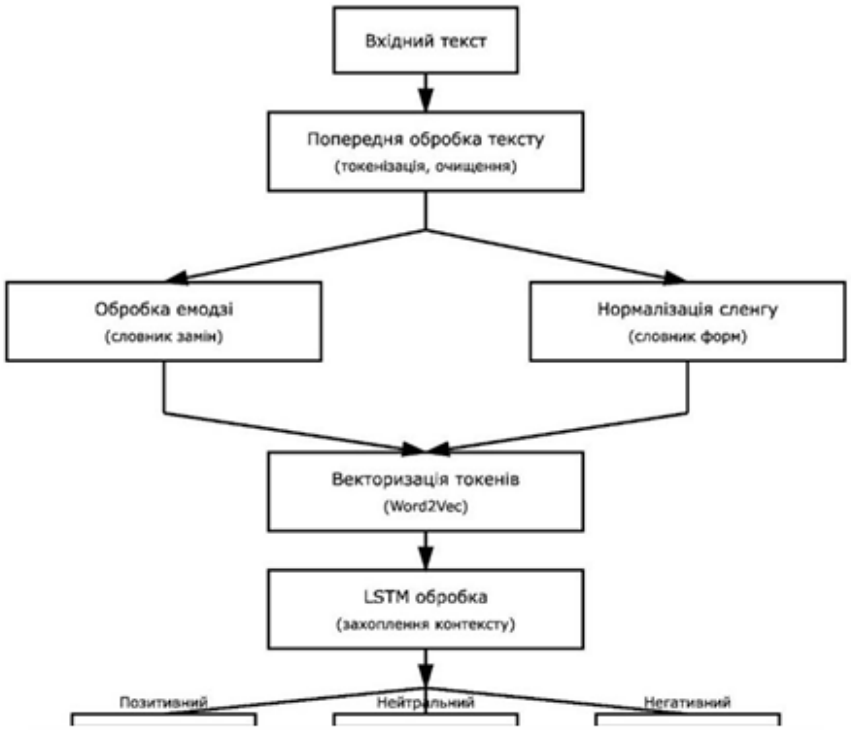


Рисунок 1 – Загальна схема методу класифікації настроїв

Архітектура нейронної мережі. Базова архітектура мережі включає послідовне з'єднання шару вкладень, LSTM шару, шару дропаут та двох повнозв'язних шарів (рисунок 2). Шар вкладень перетворює токени у векторні представлення розмірністю 300, ініціалізовані попередньо навченими векторами Word2Vec. Використовується тонке налаштування вкладень під час навчання для адаптації до специфіки задачі.

LSTM шар є основним компонентом архітектури і має 128 прихованих одиниць. Шар використовує три вентилі для контролю потоку інформації: вентиль забування визначає, яку частину попереднього стану забути; вентиль входу контролює додавання нової інформації; вентиль виходу формує прихований стан.

Після LSTM шару розташований шар дропаут з коефіцієнтом 0.5 для зменшення перенавчання. Перший повнозв'язний шар містить 512 нейронів з активацією ReLU, другий – 256 нейронів також з ReLU. Обидва повнозв'язні шари використовують дропаут 0.5. Вихідний шар має три нейрони з функцією активації софтмакс для формування розподілу ймовірностей по класах настроїв.

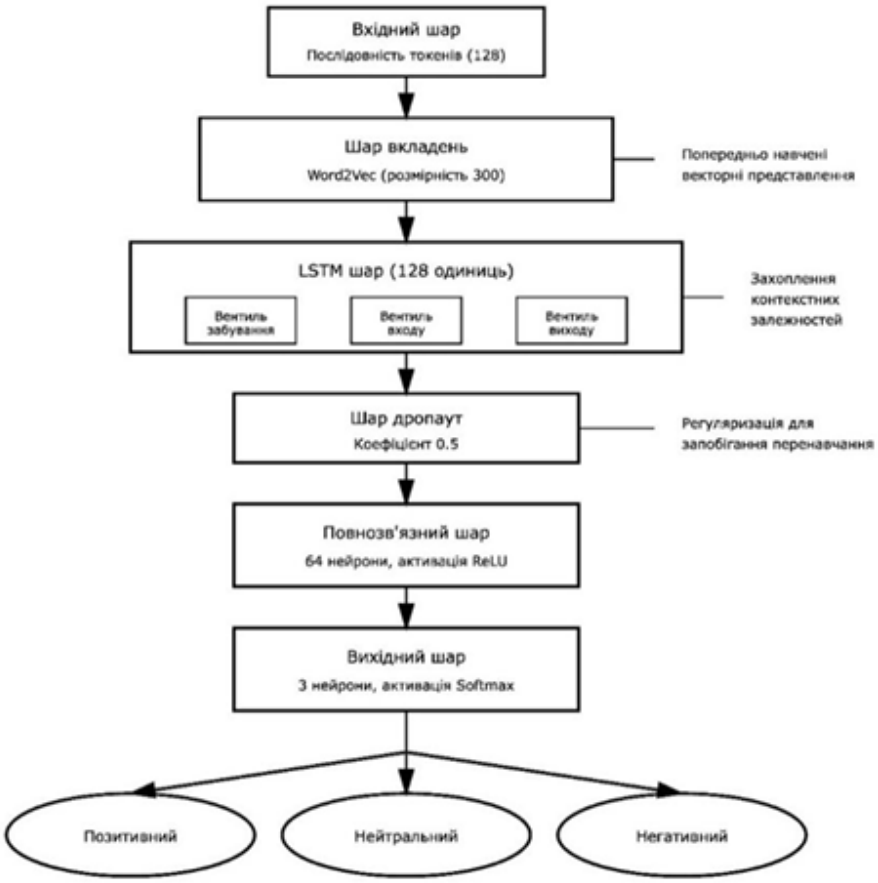


Рисунок 2 – Архітектура моделі класифікації настроїв

Отже, розроблений метод класифікації настроїв на основі рекурентних нейронних мереж LSTM забезпечує точність 89.4%. Модифікація з обробкою емодзі та сленгу через спеціальні словники покращує точність на 1.8% без збільшення обчислювальної складності. Метод є придатним для практичного застосування в системах моніторингу соціальних мереж та аналізу відгуків користувачів.

Перелік посилань

1. Murat Başal. Natural Language Processing for Sentiment Analysis in Social Media Marketing. Economics World. 2025. Vol. 12, No. 1. URL: https://doi.org/10.17265/2328-7144/2025.01.004.
2. Nikhil N., Pratik P., Rutuja P., Rutuja P., Suyash W. Sentiment Analysis Using Natural Language Processing. International Journal of Scientific Development and Research. 2022. Vol. 7, No. 5. Pp. 458–462.
3. Shad R., Potter K., Gracias A. Natural Language Processing (NLP) for Sentiment Analysis: A Comparative Study of Machine Learning Algorithms. International Journal of Artificial Intelligence and Machine Learning. 2025. Vol. 5, No. 1. Pp. 58–69. URL: https://doi.org/10.51483/IJAIML.5.1.2025.58-69.
4. Muthukumar P., Ibrahim M. A. S. Multimodal Social Media Sentiment Analysis. Stanford CS224N Final Project. 2024. Pp. 1–12.

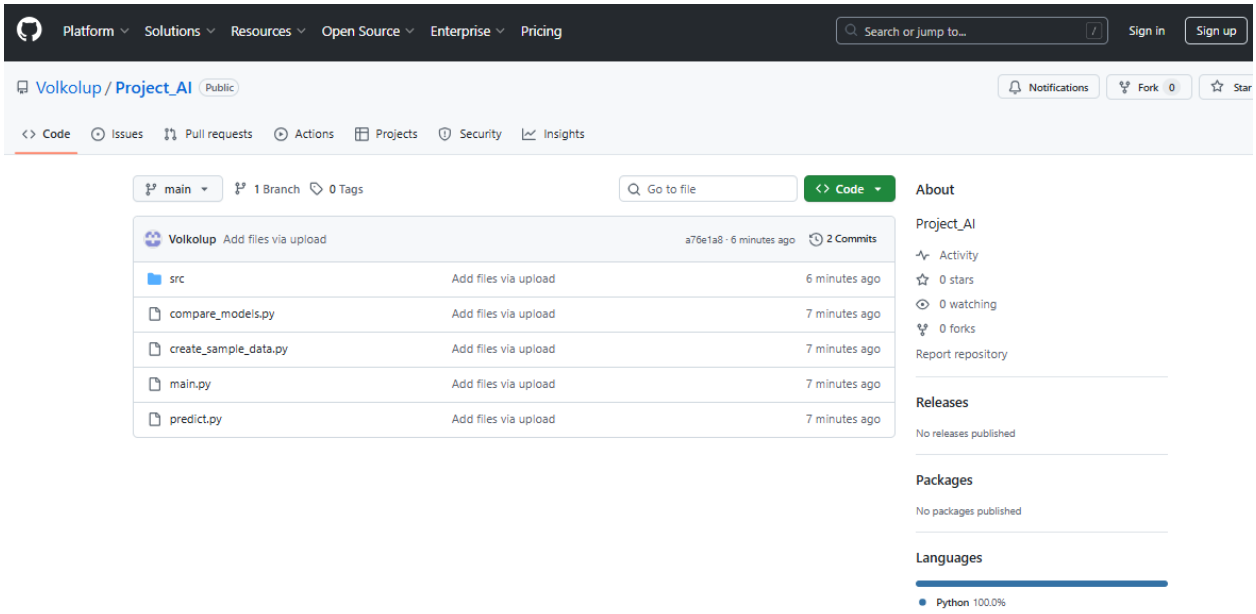
Додаток Б

Програмний код посилання на GitHub-репозиторій, структура проєкту та опис основних папок і файлів

Посилання на репозиторій на GitHub:

https://github.com/Volkolup/Project_AI

Вигляд сторінки репозиторію



text_cleaner.py - очищення тексту від URL, згадок користувачів та спеціальних символів

emoji_processor.py - обробка емодзі з конвертацією у текстові описи емоцій

slang_normalizer.py - нормалізація сленгових виразів до стандартних форм

tokenizer.py - токенізація тексту, видалення стоп-слів та створення послідовностей

Модулі машинного навчання (src/model/)

architecture.py - реалізація архітектури LSTM з шарами вкладень та дропаутом

trainer.py - алгоритми навчання моделі з callbacks та збереженням контрольних точок

Утиліти та допоміжні функції (src/evaluation/)

metrics.py - обчислення Accuracy, Precision, Recall, F1-Score для оцінки якості класифікації

Конфігурація (src/utils/)

config.py - управління гіперпараметрами моделі та шляхами до даних

Виконавчі скрипти

main.py - основний скрипт для навчання LSTM моделі з повним пайплайном обробки

compare_models.py - порівняння ефективності різних методів класифікації

predict.py - класифікація настрою для нових текстових повідомлень

create_sample_data.py - генерація тестового датасету з прикладами

Конфігурація системи

config.py - параметри LSTM-моделі, навчання та шляхи до даних

requirements.txt - залежності Python для роботи системи

Додаток В

Презентаційний матеріал

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

МЕТОД КЛАСИФІКАЦІЇ НАСТРОЇВ У СОЦІАЛЬНИХ МЕРЕЖАХ З ВИКОРИСТАННЯМ ОБРОБКИ ПРИРОДНОЇ МОВИ

Виконав:
студент 2 курсу, групи КНм-24-1
Богдан ВОЛКОЛУП
Керівник:
к.т.н., доцент кафедри КН
Олександр ПАСІЧНИК

2

Актуальність

Актуальність роботи обумовлена зростаючою потребою у автоматизованому аналізі великих обсягів текстової інформації з соціальних платформ із застосуванням сучасних технологій штучного інтелекту. Традиційні методи аналізу настроїв, які базуються на лексичних словниках та простих статистичних підходах, є недостатньо ефективними, потребують значних ресурсів для підтримки, схильні до помилок при роботі з неформальною мовою і не враховують контекстуальні особливості онлайн-комунікації.

Сучасні досягнення дозволяють суттєво підвищити якість і точність визначення емоційного забарвлення текстів, автоматизуючи процеси та забезпечуючи здатність моделей адаптуватися до специфіки різних соціальних платформ. Це сприяє кращому розумінню суспільних настроїв, підвищенню ефективності моніторингу репутації брендів, поліпшенню якості обслуговування клієнтів та оптимізації маркетингових стратегій на основі аналізу відгуків користувачів.

Мета і задачі роботи

Мета роботи полягає у підвищенні точності класифікації емоційного забарвлення текстових повідомлень у соціальних мережах шляхом розробки методу з використанням рекурентних нейронних мереж LSTM та спеціалізованої обробки специфічних елементів онлайн-комунікації.

Задачі дослідження:

– провести аналіз існуючих методів та підходів до класифікації настроїв у текстах з використанням методів машинного та глибокого навчання;

– розробити метод класифікації емоційного забарвлення з використанням рекурентних нейронних мереж на основі архітектури LSTM з модифікаціями для обробки емодзі та сленгу;

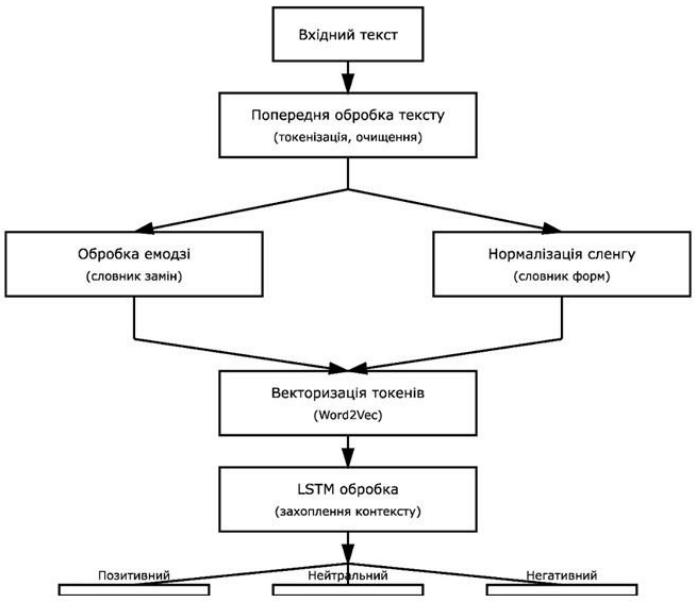
– здійснити програмну реалізацію методу класифікації емоційного забарвлення текстових повідомлень у соціальних мережах;

– провести експериментальне дослідження ефективності спроектованого методу шляхом порівняння з базовими алгоритмами класифікації та оцінки його точності на тестових даних.

Об'єкт дослідження – процес визначення емоційного забарвлення текстових повідомлень у соціальних мережах.

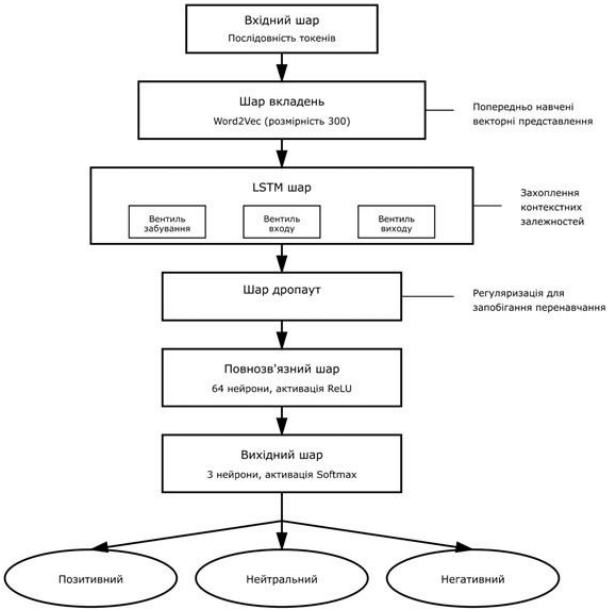
Предмет дослідження – моделі, методи та засоби класифікації настроїв на основі рекурентних нейронних мереж з обробкою емодзі та сленгових виразів.

Загальна схема методу класифікації настроїв



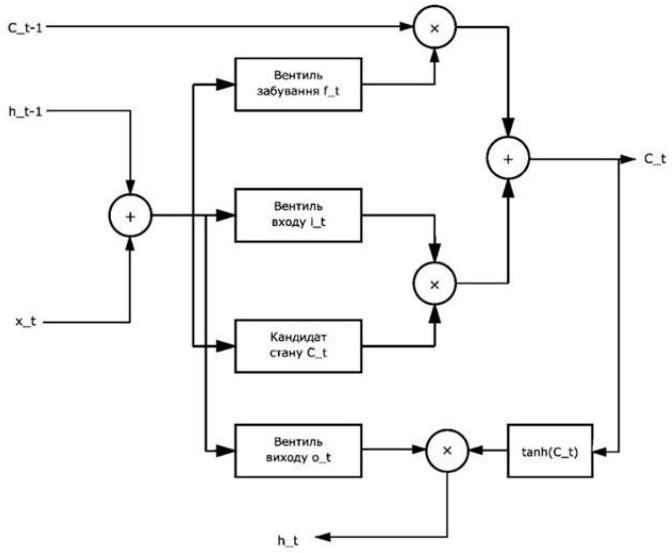
5

Архітектура моделі класифікації настроїв



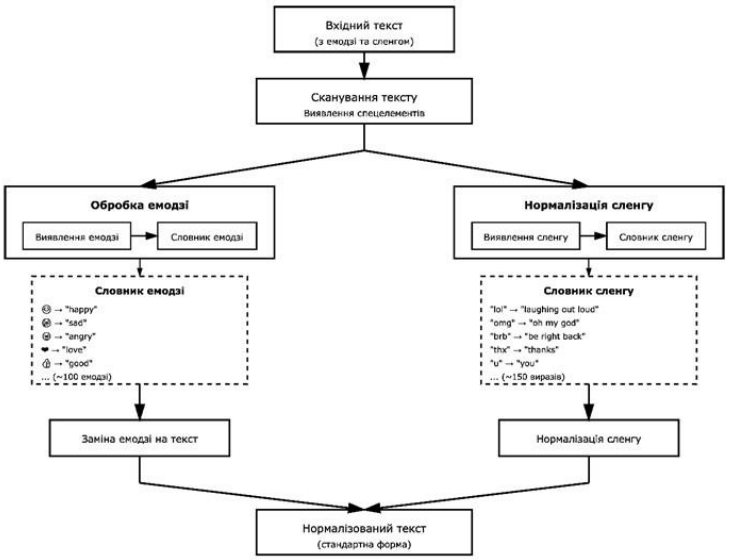
6

Структура LSTM комірки з вентилями



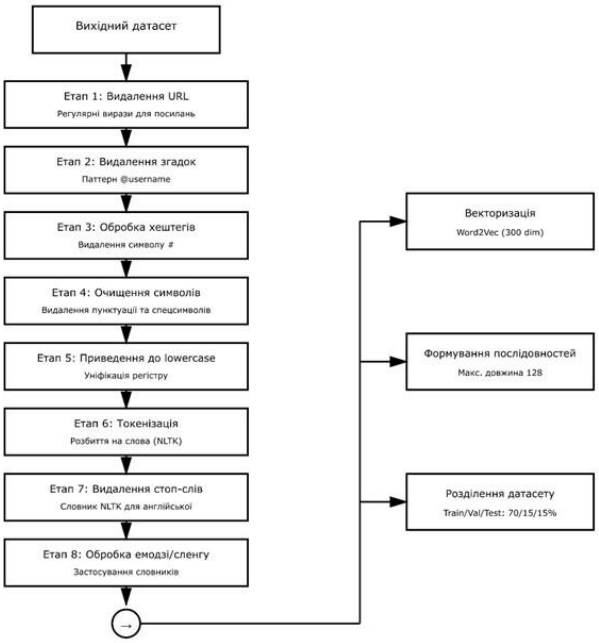
7

Схема обробки емодзі та сленгу у модифікованій моделі

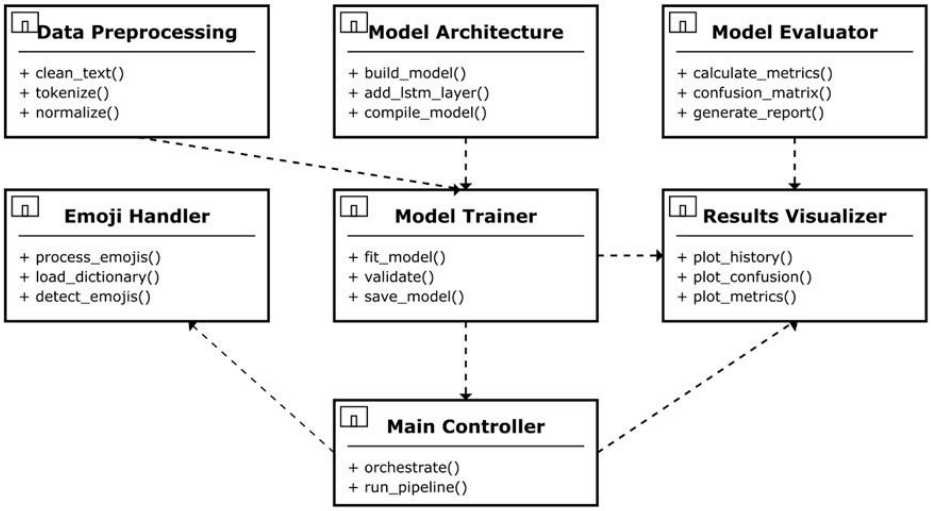


8

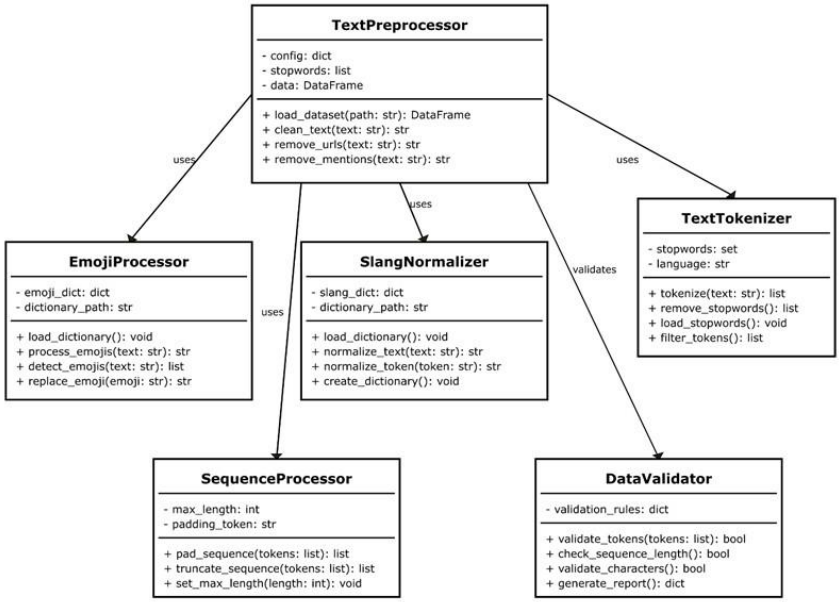
Етапи підготовки навчальних даних



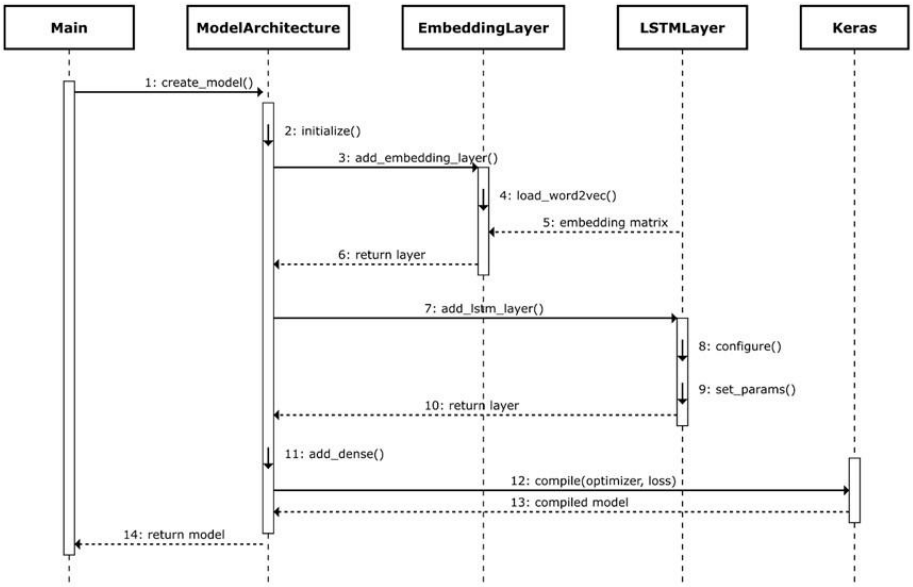
Діаграма компонентів системи класифікації



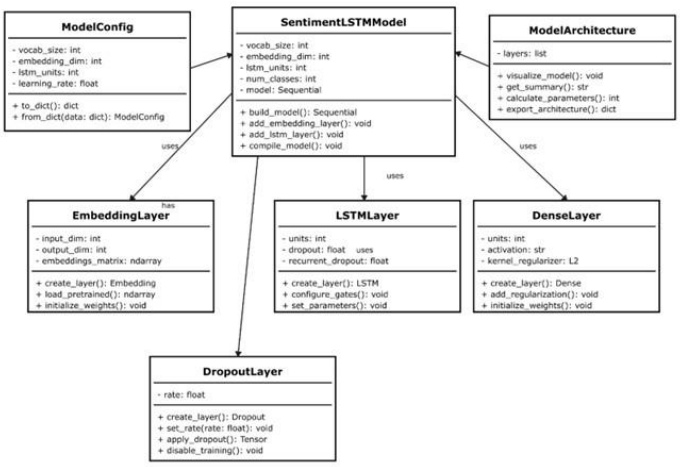
Діаграма класів модуля препроцесингу



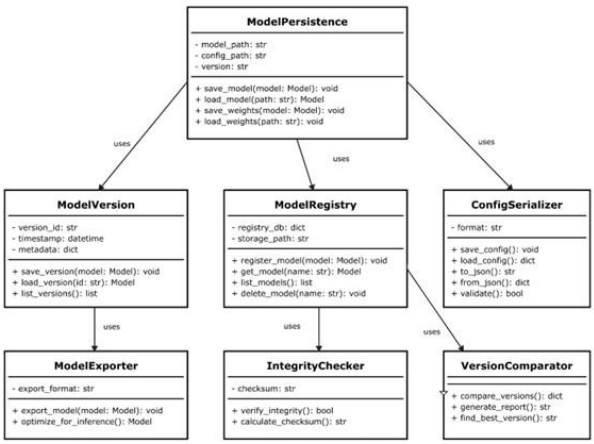
Діаграма послідовності побудови моделі



Діаграма класів модуля архітектури моделі

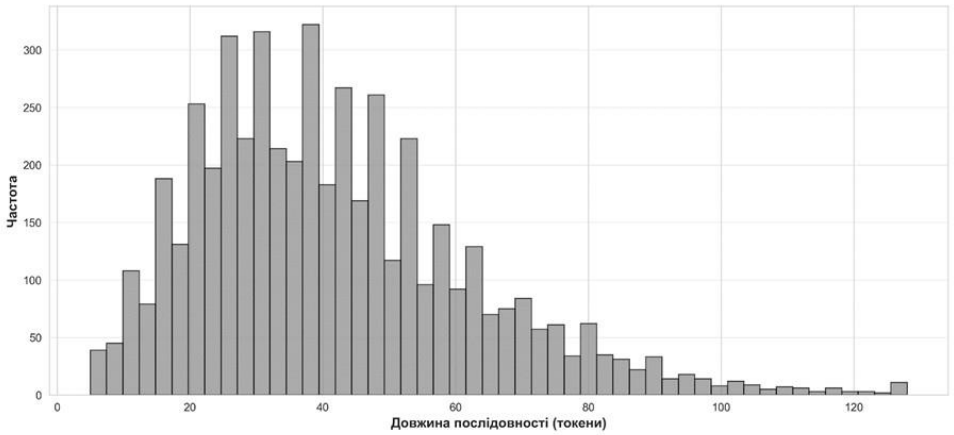


Діаграма класів модуля збереження



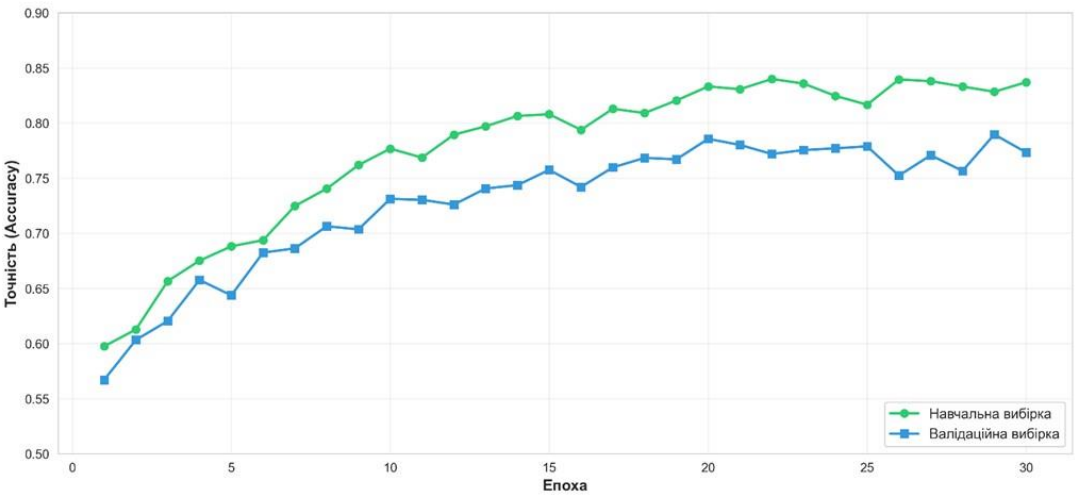
13

Розподіл довжин послідовностей токенів у датасеті



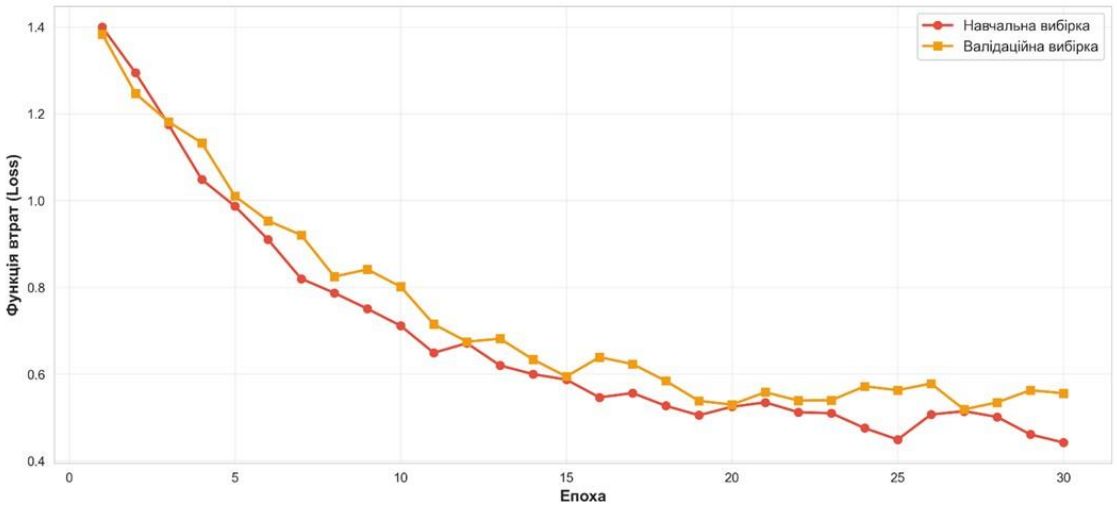
14

Крива навчання моделі LSTM за метрикою Accuracy



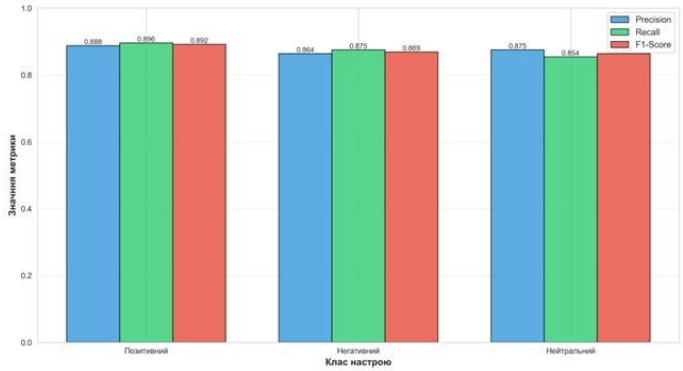
15

Крива функції втрат під час навчання

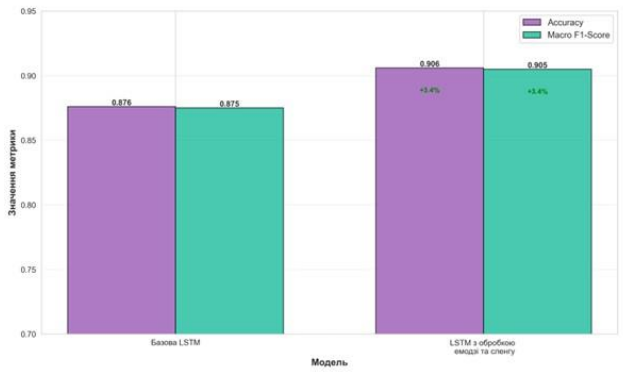


16

Порівняння метрик для різних класів настроїв



Порівняння базової та модифікованої моделі



17

Висновки

Основні результати дослідження полягають у наступному:

– проведено аналіз існуючих методів та підходів до класифікації настроїв у текстах з використанням методів машинного та глибокого навчання;

– розроблено метод класифікації емоційного забарвлення з використанням рекурентних нейронних мереж на основі архітектури LSTM з модифікаціями для обробки емодзі та сленгу;

– розроблено програму за методом класифікації емоційного забарвлення текстових повідомлень у соціальних мережах;

– проведено експериментальне дослідження ефективності спроектованого методу шляхом порівняння з базовими алгоритмами класифікації та оцінки його точності на тестових даних.

18

ДЯКУЮ ЗА УВАГУ!

Fri Dec 12 12:44:17 EET 2025, Петровський Сергій Степанович, Хмельницький національний університет, ХНУ

Anti-Plagiarism (UA) v-15.281 Educational

The maximum coincidence with one document 1.0%

Dictionaries check: en_US, ru_RU, ua_UA. **Errors in the documents: 12%**

	Document		Sum coincidence on the DB	
ID: 252661 Title: КВАЛІФІКАЦІЙНА РОБОТА на тему Метод класифікації настроїв у соціальних мережах з використанням обробки природної мови Added in a DB: 2025-12-12 Authors: Богдан ВОЛКОЛУП Heads: Олександр ПАСІЧНИК Consultants: Opponents:	Symbols	Lexemes	Symbols	Lexemes
	117266	1831	1610 (1%)	32 (2%)

Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Богдан ВОЛКОЛУП

Співавтор:

Назва: КВАЛІФІКАЦІЙНА РОБОТА на тему Метод класифікації настроїв у соціальних мережах з використанням обробки природної мови

Науковий керівник: Олександр ПАСІЧНИК, к.т.н., доцент

Підрозділ: Кафедра комп'ютерних наук

Коефіцієнт подібності 1:1.7%

Коефіцієнт подібності 2:0.4%

Мікропробіли: 0

Заміна букв: 3

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2025-12-11 21:43:11.0

Після аналізу Звіту подібності констатую наступне:

☒ **Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.**

☐ **Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.**

☐ **Виявлено запозичення і плагіат або навмисні текстові спотворення** (маніпуляції), як **передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.**

Обгрунтування:

2025-12-12

Дата

експерт Петровський С. С.

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНИХ НАУК

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи *Метод класифікації настроїв у соціальних мережах з використанням обробки природної мови*
Автор *студент групи КНм-24-1 Богдан ВОЛКОЛУП*
Освітня програма *Комп'ютерні науки*
Рівень вищої освіти *другий (магістерський)*
Спеціальність *122 – Комп'ютерні науки*
Науковий керівник: *к.т.н., доцент каф. комп'ютерних наук Олександр ПАСІЧНИК*

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмними засобами комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	***відповідає***
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	***відсутні***

Підтвердження:

Запозичення, виявлені в роботі Богдана ВОЛКОЛУПА, не є плагіатом, оскільки: запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; усі запозичення фрагментарні; до запозичень входять фрагменти, які не мають авторства і містять поширені конструкції та загальновідомі терміни, скорочення. Рівень подібності не перевищує допустимої межі. Таким чином, робота є законною та приймається до захисту.

Обсяг запозичень, визначений системами виявлення збігів/ідентичності/схожості:
- за системою Anti-Plagiarism: 1%;
- за системою StrikePlagiarism КП1: 1,69%, КП2: 0,4%.

15.12.2025

Завідувач кафедри _____ *Олександр БАРМАК*

Гарант освітньої програми _____ *Руслан БАГРІЙ*

Керівник кваліфікаційної роботи _____ *Олександр ПАСІЧНИК*

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ

Кафедра комп'ютерних наук

ВІДГУК НАУКОВОГО КЕРІВНИКА

на кваліфікаційну роботу магістра

студента *КНм-24-1 Богдана ВОЛКОЛУПА*

за темою *Метод класифікації настроїв у соціальних мережах з використанням обробки природної мови*

1. Актуальність теми

Актуальність роботи визначається зростаючою потребою у автоматизованому аналізі великих масивів текстової інформації з платформ соціальних медіа з застосуванням сучасних технологій штучного інтелекту. Використання рекурентних нейронних мереж типу LSTM у поєднанні зі спеціалізованими підходами до обробки емодзі та сленгу дозволяє суттєво підвищити якість визначення емоційного забарвлення текстів.

2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки

Дослідження повністю відповідає предметній області спеціальності 122 «Комп'ютерні науки», оскільки ґрунтується на застосуванні методів глибокого навчання, зокрема рекурентних нейронних мереж архітектури LSTM, та сучасних підходів до обробки природної мови. Робота передбачає використання алгоритмів класифікації, методів векторизації тексту, технік попередньої обробки даних, регуляризації моделей, статистичного аналізу та експериментального тестування.

3. Професійні та особистісні якості

Богдан ВОЛКОЛУП продемонстрував високий рівень професійної підготовки в області комп'ютерних наук та обробки природної мови. Він систематично і якісно виконував завдання з розробки методу класифікації емоційного забарвлення текстових повідомлень. Студент виявив глибоке розуміння архітектур нейронних мереж, принципів роботи з послідовними даними, наполегливість у проведенні експериментальних досліджень та здатність до критичного аналізу отриманих результатів.

4. Ступінь самостійності під час виконання кваліфікаційної роботи

При виконанні магістерської роботи студент виявив значний рівень самостійності. Він запропонував удосконалення методу класифікації настроїв шляхом інтеграції додаткового етапу попередньої обробки. Це включало використання спеціалізованих словників для трансформації емодзі у текстові описи емоцій та нормалізації сленгових виразів. Студент самостійно провів аналіз наукової літератури,

спроектував архітектуру моделі на основі LSTM, реалізував програмне забезпечення з модульною структурою.

5. Наукова новизна та оригінальність запропонованих підходів

Удосконалено метод класифікації настроїв у текстах соціальних мереж. Він відрізняється від існуючих використанням рекурентних нейронних мереж типу LSTM з додатковим етапом попередньої обробки. Цей етап включає створення словників для заміни емодзі на текстові описи емоцій та нормалізацію сленгових виразів.

6. Ступінь оволодіння методами дослідження

Студент продемонстрував ґрунтовне розуміння та вміле застосування методів обробки природної мови та глибокого навчання. Зокрема, він використав:рекурентні нейронні мережі, техніки векторизації тексту, методи токенізації та нормалізації даних. Це свідчить про його високий рівень оволодіння сучасними методами дослідження в галузі обробки природної мови та машинного навчання.

7. Повнота та якість розкриття теми роботи

Тема розкрита всебічно у магістерській роботі. Робота вирізняється логічною структурою, детальним аналізом існуючих підходів до класифікації настроїв у текстах. Також наявний ретельний опис розробленого методу на основі LSTM з модифікаціями для обробки специфічних елементів онлайн-комунікації.

8. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу

Магістерська робота характеризується чіткою логічною структурою, послідовним викладенням матеріалу від аналізу проблематики класифікації настроїв та існуючих підходів до розробки та експериментальної перевірки власного методу. Висновки аргументовані, підкріплені детальним статистичним аналізом з використанням метрик.

9. Можливість практичного застосування кваліфікаційної роботи, окремих її частин

Розроблений метод класифікації настроїв у соціальних мережах має широкі можливості практичного застосування в галузях автоматизованого моніторингу репутації брендів, аналізу відгуків клієнтів, виявлення кризових ситуацій в онлайн-спільнотах. Метод може бути інтегрований у сучасні системи моніторингу соціальних медіа та платформи аналізу громадської думки.

10. Висновок про можливість допуску кваліфікаційної роботи до захисту, на яку оцінку заслуговує робота

Враховуючи належний рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «відмінно».

Керівник _____ к.т.н., доцент каф. КН Олександр ПАСІЧНИК

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МОН УКРАЇНИ

Кафедра комп'ютерних наук

ВІДГУК ОПОНЕНТА

на кваліфікаційну роботу магістра

студента *гр. КНм-24-1 Богдана ВОЛКОЛУПА*

за темою *Метод класифікації настроїв у соціальних мережах з використанням обробки природної мови*

1. Актуальність обраної теми

Актуальність обраної теми зумовлена зростаючою потребою бізнесу та дослідників у автоматизованих інструментах аналізу великих обсягів текстової інформації з соціальних платформ для розуміння суспільних настроїв, моніторингу репутації брендів та оптимізації маркетингових стратегій. Застосування методів глибокого навчання, зокрема рекурентних нейронних мереж та спеціалізованих технік обробки неформальної мови соціальних медіа, дозволяє значно підвищити ефективність визначення емоційного забарвлення текстів.

2. Відповідність роботи предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до наукових робіт

Магістерська робота повністю відповідає предметній області спеціальності 122 Комп'ютерні науки, оскільки базується на застосуванні методів глибокого навчання, обробки природної мови та створенні інтелектуальних систем для вирішення задачі класифікації емоційного забарвлення текстів.

3. Повнота розкриття мети та завдань дослідження

Мета та завдання дослідження розкриті повно та послідовно. Автор чітко формулює мету роботи – підвищення точності класифікації емоційного забарвлення текстових повідомлень у соціальних мережах шляхом розробки методу з використанням рекурентних нейронних мереж LSTM та спеціалізованої обробки специфічних елементів онлайн-комунікації. Для досягнення мети вирішуються поставлені завдання, що включають аналіз існуючих методів класифікації настроїв, розробку методу з модифікаціями для обробки емодзі та сленгу, програмну реалізацію та експериментальне дослідження на реальних даних.

4. Наявність наукової новизни

Наукова новизна роботи полягає в удосконаленні методу класифікації настроїв у текстах соціальних мереж, який відрізняється від існуючих використанням рекурентних

нейронних мереж типу LSTM з додатковим етапом попередньої обробки, що включає створення словників для заміни емодзі на текстові описи емоцій.

5. Зміст кожного розділу роботи

Робота містить чотири розділи та структурована логічно. Перший розділ присвячений дослідженню проблеми класифікації настроїв у соціальних мережах, характеристиці задачі, аналізу існуючих публікацій та наукових підходів, а також огляду архітектур, методів та моделей обробки природної мови. Другий розділ описує розроблений метод класифікації з концепцією, архітектурою моделі на основі LSTM, модифікаціями для обробки емодзі та сленгу, формуванням навчальних даних та критеріями оцінювання. Третій розділ містить програмну реалізацію методу з описом технологічного стеку, структури рішення та основних компонентів. Четвертий розділ представляє експериментальне дослідження, оцінювання якості на тестовій вибірці.

6. Ступінь розкриття теми роботи

Тема роботи розкрита всебічно та глибоко. Автор детально аналізує проблематику класифікації настроїв у текстах соціальних мереж, розглядає існуючі методи машинного та глибокого навчання для цієї задачі, обґрунтовує необхідність розробки удосконаленого методу. Описано запропонований метод на основі архітектури LSTM з інтегрованими словниками емодзі та сленгу, наведено архітектуру моделі з математичним описом роботи вентилів LSTM.

7. Якість оформлення кваліфікаційної роботи

Якість оформлення кваліфікаційної роботи відповідає встановленим академічним стандартам. Текст написаний грамотною українською мовою з дотриманням наукового стилю.

8. Недоліки кваліфікаційної роботи

Корисно було б розширити експериментальне дослідження на мультимовні датасети для оцінки можливості переносу методу на інші мови.

9. Загальний висновок (допускається чи не допускається до захисту), якої оцінки заслуговує кваліфікаційна робота

Враховуючи рівень виконання та забезпечення усіх необхідних вимог, наявність наукової новизни отриманих результатів, якість проведених експериментальних досліджень, апробацію результатів на науковій конференції, робота може бути допущена до захисту. Рекомендована оцінка – «відмінно».

Опонент д.т.н., проф. каф АСУТП ________

Мерочник В.В.