ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ШТУЧНОГО ІНТЕЛЕКТУ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «РОЗРОБКА АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ ДЛЯ АНАЛІЗУ СЕНТИМЕНТУ ТА ЕМОЦІЙ В INSTAGRAM»

на здобуття освітнього ступеня бакалавра
зі спеціальності 122 Комп'ютерні науки

освітньо-професійної програми Штучний інтелект

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Дмитро РУЛЬ

Виконав:
здобувач вищої освіти Дмитро РУЛЬ
група ШІД-41

Керівник: Антон ШАНТИР
к.т.н.

Рецензент:
_____

Київ 2024

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Штучного інтелекту

Ступінь вищої освіти Бакалавр

Спеціальність 122 Комп'ютерні науки

Освітньо-професійна програма Штучний інтелект

ЗАТВЕРДЖУЮ
Завідувач кафедрою Штучного інтелекту

_____ Ольга ЗІНЧЕНКО
«_____» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Руль Дмитро Володимирович

1. Тема кваліфікаційної роботи: «РОЗРОБКА АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ ДЛЯ АНАЛІЗУ СЕНТИМЕНТУ ТА ЕМОЦІЙ В INSTAGRAM»
керівник кваліфікаційної роботи Антон ШАНТИР д.т.н.,
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «27» 02.2024р. № 36

2. Строк подання кваліфікаційної роботи «31» травня 2024р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, специфікації та документація з Python, RNN, CNN, BERT та JSON, вимоги до вимоги до системи.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
   1. Аналіз існуючих методів і моделей для аналізу сентименту та емоцій.
   2. Розробка алгоритму аналізу тональності коментарів в Instagram.
   3. Навчання і тестування моделей на зібраних даних.
   4. Оцінка ефективності запропонованих методів.

5. Перелік графічного матеріалу: *презентація*
6. Дата видачі завдання «27» лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	27.02-05.11.23	
2	Вибір технологій для реалізації системи	05.11-12.11.23	
3	Огляд сучасних технологій та методів для сентимент аналізу текстів	13.11-19.11.23	
4	Розробка архітектура системи аналізу тональності коментарів	20.11-25.11.23	
5	Дослідження інтеграції з попередньо навченими моделями	27.11-03.12.23	
6	Розробка системи аналізу тональності Додаток в	04.12-10.12.23	
7	Оформлення роботи: вступ, висновки, реферат	11.12-20.12.23	
8	Розробка демонстраційних матеріалів	21.12-31.05.24	

Здобувач вищої освіти _____ Дмитро РУЛЬ

Керівник
кваліфікаційної роботи _____ Антон ШАНТИР

РЕФЕРАТ

Текстова частина бакалаврської роботи: 68 стор., 6 табл., 12 рис., 24 джерела.

Мета роботи - розробка та вдосконалення алгоритмів машинного навчання для автоматичного аналізу сентименту та емоцій в Instagram.

Об'єкт дослідження – це процес аналізу коментарів, що публікуються користувачами під постами в соціальній мережі Instagram.

Предмет дослідження - система алгоритмів машинного навчання для аналізу сентименту та емоцій у текстових даних Instagram.

Короткий зміст роботи: Методи машинного навчання, обробка природної мови (NLP), методи попередньої обробки даних, такі як токенізація, лемматизація, а також спеціалізовані підходи для аналізу текстового контенту. Визначено підхід до аналізу тональності коментарів. Здійснено розробку системи для аналізу тональності коментарів з постів Instagram. На основі результатів виконаних досліджень розроблено систему для аналізу тональності коментарів з постів Instagram.

Впровадження розробленої системи (методики) дозволяє аналізувати коментарі з постів соціальної мережі Instagram.

КЛЮЧОВІ СЛОВА: СЕНТИМЕНТ АНАЛІЗ, АНАЛІЗ ТОНАЛЬНОСТІ ТЕКСТУ, BERT, TOBERTA.

ЗМІСТ

ВСТУП

У сучасному світі соціальні мережі стали невід'ємною частиною життя мільйонів людей, що створює величезний обсяг даних для аналізу. Instagram, як одна з найпопулярніших платформ, генерує величезні масиви текстової та візуальної інформації, що містить різноманітні емоційні і сентиментальні відтінки. Аналіз цих даних може надати важливу інформацію для бізнесу, маркетингу, соціології та психології. Актуальність теми обумовлена необхідністю розробки ефективних алгоритмів машинного навчання для автоматичного визначення та аналізу сентименту і емоцій у постах та коментарях користувачів Instagram.

Проблема аналізу сентименту і емоцій у текстових даних активно досліджується науковцями з різних країн. Однак, більшість існуючих досліджень зосереджена на англомовних текстах і не враховує специфіку візуального контенту Instagram. Відтак, є потреба в розробці методик, що будуть враховувати багатомовність та мультимедійність даних соціальних мереж.

Джерельна база дослідження включає пости та коментарі користувачів Instagram, які будуть зібрані за допомогою API платформи. Додатково використовуватимуться корпуси текстів для навчання моделей машинного навчання, а також набори даних для аналізу емоційного забарвлення текстів, створені на основі відкритих джерел.

Об'єктом дослідження є процес аналізу коментарів, що публікуються користувачами під постами в соціальній мережі Instagram.

Предметом дослідження є система алгоритмів машинного навчання для аналізу сентименту та емоцій у текстових даних Instagram.

Метою роботи є розробка та вдосконалення алгоритмів машинного навчання для автоматичного аналізу сентименту та емоцій в Instagram. Завданнями роботи є:

1. Аналіз існуючих методів і моделей для аналізу сентименту та емоцій.
2. Розробка алгоритму аналізу тональності коментарів в Instagram.
3. Навчання і тестування моделей на зібраних даних.
4. Оцінка ефективності запропонованих методів.

Методика дослідження

У дослідженні використовуються методи машинного навчання, включаючи наглядове навчання, глибоке навчання та обробку природної мови (NLP). Застосовуватимуться методи попередньої обробки даних, такі як токенізація, лемматизація, а також спеціалізовані підходи для аналізу текстового контенту.

Наукова новизна роботи

Наукова новизна роботи полягає у розробці методик та алгоритмів для аналізу сентименту і емоцій, які враховують особливості даних Instagram, такі як багатомовність та мультимедійність. Це дозволить підвищити точність і ефективність аналізу емоційного стану користувачів на основі їхніх публікацій.

Практична значущість результатів дослідження

Результати дослідження можуть бути використані в різних сферах, включаючи маркетинг, для виявлення емоційного ставлення споживачів до продуктів та послуг, а також в соціологічних та психологічних дослідженнях для вивчення емоційного стану великих груп людей. Це також може сприяти розробці інструментів для моніторингу громадської думки та прогнозування соціальних тенденцій.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд та аналіз машинного навчання та обробки природної мови (NLP)

1.1.1 Кероване навчання (Supervised Learning)

Кероване навчання є підходом до машинного навчання, де модель навчається на основі підписаних даних (тобто даних, де вхідні значення мають відповідні мітки) [16]. Мета полягає у створенні функції, яка може передбачати вихідні значення для нових, непідписаних даних.

Переваги:

- Висока точність - моделі можуть досягати високої точності завдяки використанню великих обсягів підписаних даних.
- Передбачуваність - модель може бути добре налаштована на конкретне завдання, що забезпечує передбачуваність результатів.
- Придатність для численних застосувань - використовується у різних галузях, включаючи класифікацію, регресію та розпізнавання образів.

Недоліки:

- Потреба у великих обсягах підписаних даних - збір та маркування даних може бути дорогим та трудомістким процесом.
- Можливість перенавчання (overfitting) - модель може надто добре підлаштуватися під навчальні дані, що знижує її здатність до узагальнення нових даних.
- Часові та ресурсні витрати - навчання моделей може потребувати значних обчислювальних ресурсів та часу.

1.1.2 Некероване навчання (Unsupervised Learning)

Некероване навчання працює з даними без попередньої маркування [17]. Метою є виявлення прихованих структур, патернів або групувань у даних.

Переваги:

- Дані не потребують попереднього маркування, що знижує витрати на їх підготовку.
- Допомагає виявити невідомі раніше патерни та структури у даних.
- Використовується для кластеризації, зменшення розмірності, виявлення аномалій та ін.

Недоліки:

- Некероване навчання може бути менш точним у порівнянні з наглядовим навчанням.
- Результати можуть бути важкими для інтерпретації та оцінки.
- Параметри, такі як кількість кластерів у кластеризації, можуть бути важкими для визначення без попереднього знання даних.

1.1.3 Напівкероване навчання (Semi-supervised Learning)

Напівкероване навчання є підходом, що поєднує наглядове та ненаглядове навчання[18]. Використовує невелику кількість підписаних даних разом з великою кількістю непідписаних даних для покращення точності моделей.

Переваги:

- Використання меншої кількості підписаних даних знижує витрати на їх підготовку.
- Може досягати кращих результатів у порівнянні з виключно ненаглядовим навчанням, використовуючи інформацію з підписаних даних.
- Часто дані у реальних застосуваннях містять обмежену кількість підписаних прикладів, тому напівнаглядове навчання є практичним підходом.

Недоліки:

- Поєднання керованих та некерованих методів може призводити до складності моделі.
- Якість та розподіл підписаних та непідписаних даних можуть значно впливати на результати.

- Вимагає ретельної обробки та підготовки даних для досягнення оптимальних результатів.

1.1.4 Навчання з підкріпленням (Reinforcement Learning)

Навчання з підкріпленням є підходом, де агент навчається взаємодіяти з середовищем, отримуючи винагороди або покарання за свої дії[19]. Мета полягає у максимізації кумулятивної винагороди.

Переваги:

- Агент може адаптувати свою поведінку до змін у середовищі.
- Ефективний у ситуаціях, де важко визначити чіткі правила або де середовище постійно змінюється.
- Може вирішувати складні задачі, такі як гра в шахи або керування роботами.

Недоліки:

- Навчання агентів може вимагати значних обчислювальних ресурсів.
- Може бути складним у реалізації, особливо у середовищах з великою кількістю можливих станів та дій.
- Агент може потребувати багатьох ітерацій для досягнення оптимальної поведінки.

Таблиця 1.1

Порівняння методів машинного навчання

Метод	**Переваги**	**Недоліки**
Кероване навчання	Висока точність, передбачуваність, придатність для численних застосувань	Потреба у великих обсягах підписаних даних, можливість перенавчання, часові та ресурсні витрати
Некероване навчання	Відсутність потреби у підписаних даних, відкриття нових знань, застосовність до різних типів завдань	Менша точність, складність інтерпретації результатів, необхідність вибору параметрів

Продовження Таблиці 1.1

Метод	Переваги	Недоліки
Напівкероване навчання	Зниження витрат на маркування, покращення точності, застосовність у реальних сценаріях	Складність моделі, чутливість до якості даних, необхідність ретельної обробки
Навчання з підкріпленням	Адаптивність, використання в складних середовищах, рішення складних задач	Високі обчислювальні витрати, складність реалізації, потреба у великій кількості ітерацій

Попередня обробка тексту є важливим етапом в обробці природної мови (NLP), що включає різноманітні техніки для очищення та підготовки текстових даних перед їх аналізом або подальшою обробкою. Основною метою попередньої обробки є перетворення неструктурованих текстових даних у форму, яка є більш структурованою та придатною для алгоритмів машинного навчання та аналізу.

1.1.5 Токенізація (Tokenization)

Токенізація — це процес розбиття тексту на окремі частини, звані токенами, які можуть бути словами, фразами або навіть символами[1]. Токени є основними одиницями для подальшої обробки тексту.

Переваги:

- Токенізація розбиває текст на зрозумілі одиниці, що спрощує їх аналіз.
- Є базовою операцією для багатьох інших методів попередньої обробки, таких як лемматизація та стемінг.

Недоліки:

- Мови з складною морфологією або ієрогліфічними системами потребують специфічних підходів до токенізації.
- Токенізація може не враховувати контекстні відтінки, такі як омоніми.

1.1.6 Стоп-слова (Stop Words)

Стоп-слова — це загальні слова, які часто зустрічаються у тексті, але не несуть значущої смислової інформації (наприклад, "і", "на", "в")[20]. Видалення стоп-слів допомагає зменшити обсяг даних та покращити продуктивність моделей.

Переваги:

- Видалення стоп-слів зменшує розмір тексту, що спрощує його обробку.
- Зосередження на значущих словах покращує точність моделей.

Недоліки:

- Іноді стоп-слова можуть бути важливими для контексту, особливо в завданнях, де значення слова може змінюватися в залежності від його оточення.
- Некоректно складений список стоп-слів може призвести до видалення важливих слів.

1.1.7 Лематизація та стемінг (Lemmatization and Stemming)

Лематизація і стемінг — це техніки для зведення слів до їх базової або кореневої форми[21]. Лематизація зводить слово до його лемми, враховуючи контекст (наприклад, "running" до "run"), тоді як стемінг обрізає закінчення для отримання кореневої форми (наприклад, "running" до "run").

Переваги:

- Обидві техніки допомагають звести різні форми одного слова до єдиної форми, що покращує узагальнення моделей.
- Зменшення кількості різних форм слів знижує розмірність текстових даних.

Недоліки:

- Стемінг може іноді обрізати слова надто агресивно, втрачаючи важливу інформацію.
- Лематизація є складнішим процесом, що вимагає більше обчислювальних ресурсів та врахування контексту.

1.1.8 Векторизація тексту (Text Vectorization)

Векторизація тексту — це процес перетворення текстових даних у числові вектори, які можуть бути використані алгоритмами машинного навчання[22]. Основні методи включають Bag of Words (BoW) та TF-IDF.

Bag of Words (BoW) представляє текст як множину слів, без урахування їх порядку, де кожне слово є окремим елементом вектора з кількістю повторень цього слова в тексті[22].

Переваги:

- Легкий у реалізації та зрозумілий метод.
- Використовується у багатьох NLP задачах.

Недоліки:

- Не враховує порядок слів та їх контекст.
- Вектори можуть бути дуже великими, особливо для великих текстових корпусів.

TF-IDF (Term Frequency-Inverse Document Frequency) — це статистичний метод, що відображає важливість слова у документі відносно всього корпусу документів [22]. TF вимірює частоту слова в документі, тоді як IDF знижує вагу слів, які часто зустрічаються у багатьох документах.

Переваги:

- Виділяє більш значущі слова, знижуючи вагу загальних слів.
- Враховує частоту та розподіл слів у текстах.

Недоліки:

- Як і BoW, не враховує порядок слів у тексті.
- Більш складний у реалізації порівняно з BoW та потребує більше обчислювальних ресурсів.

Таблиця 1.2

Порівняння методів попередньої обробки тексту

Метод	**Переваги**	**Недоліки**
Токенізація	Спрощує аналіз, основний крок для інших методів	Різні підходи для різних мов, складність у розпізнаванні контексту
Стоп-слова	Зменшує обсяг даних, покращує продуктивність моделей	Можливі втрати важливої інформації, можливість помилок при видаленні
Лемматизація та стемінг	Покращує узагальнення, зменшує розмірність даних	Втрати точної інформації (стемінг), складність та ресурсоємність (лемматизація)
Bag of Words (BoW)	Простота реалізації, придатність для різних задач	Ігнорування контексту, висока розмірність
TF-IDF	Врахування важливості слів, більш точне відображення тексту	Ігнорування порядку слів, складність розрахунків

Ці методи попередньої обробки тексту є критично важливими для успішної реалізації задач обробки природної мови та машинного навчання. Вибір конкретного методу залежить від специфіки задачі та характеру даних.

Моделі представлення слів є важливою частиною обробки природної мови (NLP), оскільки вони дозволяють перетворювати слова у числові вектори, які можуть бути використані алгоритмами машинного навчання[2,13]. Нижче наведені три популярні моделі представлення слів: Word2Vec, GloVe та FastText.

Word2Vec — це модель представлення слів, розроблена командою Google, яка використовує нейронні мережі для навчання векторних представлень слів[13]. Існує дві основні архітектури Word2Vec: CBOW (Continuous Bag of Words) і Skip-gram.

- CBOW - передбачає поточне слово за його контекстом (навколишніми словами).
- Skip-gram - передбачає контекст за поточним словом.

Переваги:

- Вектори слів навчаються таким чином, що слова з подібними контекстами мають схожі вектори.
- Використання нейронних мереж забезпечує швидке та ефективне навчання.
- Добре працює з великими обсягами даних і може бути застосоване до різних мов.

Недоліки:

- Word2Vec розглядає кожне слово як окрему одиницю, не враховуючи внутрішню структуру слова.
- Для отримання якісних векторів потрібно багато даних для навчання.

GloVe (Global Vectors for Word Representation) — це модель представлення слів, розроблена командою Стенфордського університету, яка комбінує переваги глобальних статистичних методів та локальних контекстних методів[2,3]. GloVe навчається на статистиці співзустрічей слів у великому корпусі тексту.

Переваги:

- Використовує глобальну статистику співзустрічей слів, що дозволяє краще враховувати їх значення.
- Вектори слів є стабільними та добре відображають семантичні відношення між словами.
- Ефективний алгоритм, який забезпечує швидке навчання.

Недоліки:

- Як і Word2Vec, GloVe не враховує внутрішню морфологію слів.
- Навчання потребує великих корпусів текстів.

FastText — це модель представлення слів, розроблена компанією Facebook, яка розширює Word2Vec, враховуючи внутрішню структуру слів. FastText

представляє кожне слово як сукупність n-грам, що дозволяє краще обробляти рідкісні слова та нові слова[4].

Переваги:

- Використання n-грам дозволяє моделі враховувати морфологічні властивості слів.
- Добре працює з рідкісними словами та новими словами, яких не було у навчальному корпусі.
- Підходить для різних мов і легко адаптується до нових доменів.

Недоліки:

- Врахування n-грам робить процес навчання більш складним та ресурсоємним.
- Модель може мати великий розмір через врахування всіх можливих n-грам.

Таблиця 1.3

Порівняння моделей представлення слів

Модель	**Переваги**	**Недоліки**
Word2Vec	Врахування контексту, ефективність навчання, універсальність	Ігнорування морфології слів, потреба у великих даних
GloVe	Врахування глобальної інформації, стабільність векторів, швидке навчання	Статична природа, потреба у великих обсягах даних
FastText	Врахування морфології, обробка рідкісних слів, універсальність	Складність навчання, великий розмір моделі

Ці моделі є основними інструментами для представлення слів у числовому вигляді та використовуються у багатьох задачах NLP, включаючи класифікацію тексту, машинний переклад та інші області обробки природної мови. Вибір конкретної моделі залежить від специфіки задачі, доступних обчислювальних ресурсів та обсягу даних для навчання.

1.2 Методи класифікації тональності

Тональність тексту — це характеристика, яка вказує на емоційне забарвлення повідомлення[3,5]. Тональність може бути позитивною, негативною або нейтральною. Визначення тональності [6,9]

- Позитивна тональність: текст, який виражає позитивні емоції, задоволення, схвалення або підтримку. Наприклад, "Мені дуже сподобався цей продукт!"
- Негативна тональність: текст, який виражає негативні емоції, розчарування, критику або невдоволення. Наприклад, "Цей сервіс був жахливим."
- Нейтральна тональність: текст, який не виражає явно позитивних або негативних емоцій, є об'єктивним або інформаційним. Наприклад, "Цей продукт виготовлено з пластика."

1. Аналіз тональності в маркетингу [8]

- Відгуки клієнтів: аналіз відгуків допомагає компаніям розуміти думку клієнтів про їх продукти або послуги, ідентифікувати сильні та слабкі сторони.
- Брендовий імідж: моніторинг соціальних мереж та інших платформ дозволяє оцінити, як споживачі сприймають бренд.
- Розробка стратегій: результати аналізу тональності допомагають у формуванні маркетингових стратегій та кампаній, спрямованих на покращення взаємодії з клієнтами.

2. Аналіз тональності в соціальних мережах[8]

- Моніторинг настроїв: відстеження тональності публікацій у соціальних мережах допомагає визначити загальний настрій аудиторії та швидко реагувати на негативні тенденції.
- Кризовий менеджмент: швидке виявлення негативних тенденцій дозволяє вчасно реагувати на потенційні кризи та мінімізувати репутаційні ризики.
- Аналіз конкурентів: оцінка тональності обговорень, пов'язаних з конкурентами, допомагає визначити їхні сильні та слабкі сторони, а також розробити ефективні стратегії конкуренції.

Аналіз тональності тексту є важливим інструментом для багатьох сфер, дозволяючи отримати цінну інформацію про емоційне сприйняття текстів та реакції на них. Це сприяє прийняттю обґрунтованих рішень та розробці ефективних стратегій взаємодії з аудиторією.

1.2.1 Лексичні методи

Лексичні методи класифікації тональності базуються на використанні попередньо створених словників або лексиконів, які містять слова з відповідними тональними оцінками[2]. Ці методи не потребують великих обсягів даних для навчання, але можуть бути менш точними у порівнянні з моделями машинного навчання.

Використання словників і лексиконів є одним із найпростіших методів для аналізу тональності тексту. Основна ідея полягає в тому, щоб зіставити слова тексту з словами у словнику, які мають попередньо визначені тональні значення (позитивні, негативні або нейтральні). Оцінка тональності тексту здійснюється на основі сумарної тональності слів, які містяться у словнику.

Основні кроки цього підходу включають[2]:

1. Створення або використання існуючого лексикону, який містить слова з відповідними тональними оцінками.
2. Розбиття тексту на окремі слова або фрази.
3. Порівняння слів тексту з словами у лексиконі.
4. Сумування або обчислення середнього значення тональності знайдених слів для отримання загальної тональності тексту.

Приклади популярних словників

SentiWordNet — це анотований лексикон, заснований на WordNet, який призначає кожному слову або сенсу слова оцінку тональності (позитивну, негативну або нейтральну)[2]. Кожне слово має кілька сенсів з різними тональними оцінками, що дозволяє враховувати багатозначність слів. Використовується для аналізу тональності тексту в різних мовах, особливо в дослідженнях англійської мови.

AFINN — це лексикон для аналізу тональності, який містить близько 2500 англійських слів з оцінками тональності в діапазоні від -5 (дуже негативне) до +5 (дуже позитивне)[2]. Простий у використанні, включає тільки слова з вираженою позитивною або негативною тональністю. Підходить для швидкого аналізу тональності тексту, особливо в англійській мові.

Переваги:

1. Лексичні методи легко реалізувати і вони не вимагають великих обчислювальних ресурсів.
2. Такі методи можуть швидко аналізувати текст, оскільки вони не потребують навчання моделей.
3. Результати аналізу легко інтерпретувати, оскільки кожне слово має чітко визначену тональність.
4. Не потребують великих обсягів навчальних даних, що є перевагою у випадках, коли підписані дані важко отримати.

Недоліки:

1. Словники можуть бути обмеженими та не охоплювати всі можливі слова та фрази, що знижує точність аналізу.
2. Лексичні методи не враховують контекст, у якому використовується слово, що може призводити до неправильних оцінок.
3. Створення та підтримка актуальності лексиконів потребує значних зусиль, особливо для нових слів і сленгу.
4. Слова можуть мати різні значення залежно від контексту, і лексичні методи не завжди здатні це врахувати.

Лексичні методи є простими та швидкими інструментами для класифікації тональності тексту. Вони особливо корисні у випадках, коли немає великих обсягів підписаних даних або коли потрібен швидкий аналіз. Однак, для досягнення високої точності аналізу тональності, особливо у складних текстах з багатозначністю та контекстними залежностями, можуть знадобитися більш просунуті методи, такі як моделі машинного або глибокого навчання.

1.2.2 Методи на основі правил

Методи на основі правил передбачають створення набору чітких інструкцій, які визначають тональність тексту[25]. Ці правила можуть бути розроблені вручну і ґрунтуються на лінгвістичних знаннях та аналізі текстів.

1. Вибір ключових слів і фраз

- Позитивні слова та фрази (наприклад, "прекрасний", "відмінний", "люблю")
- Негативні слова та фрази (наприклад, "жахливий", "не задоволений", "ненавиджу")
- Нейтральні слова та фрази (наприклад, "продукт", "обслуговування", "доставка")

2. Врахування інтенсивності

- Інтенсивність слів може змінювати значення. Наприклад, слово "гарний" має меншу інтенсивність порівняно з "прекрасний".
- Використання модифікаторів для підсилення або послаблення емоційного забарвлення (наприклад, "дуже", "трохи", "абсолютно").

3. Заперечення

- Заперечення може змінювати тональність речення. Наприклад, "не поганий" має позитивну тональність, незважаючи на наявність слова "поганий".
- Важливо враховувати структуру речень, щоб правильно ідентифікувати заперечення.

4. Контекстуальні підказки

- Аналіз контекстуальних підказок, таких як послідовність слів та граматична структура речень.
- Використання правила виявлення сарказму, іронії та інших стилістичних фігур.

Приклади правил

1. Правило для позитивної тональності:

- Якщо текст містить слова зі списку позитивних, і відсутні заперечення, то тональність тексту позитивна.
- Наприклад, "Мені дуже сподобався цей продукт!" (містить "сподобався", що є позитивним словом).

2. Правило для негативної тональності:

- Якщо текст містить слова зі списку негативних, і відсутні заперечення, то тональність тексту негативна.
- Наприклад, "Цей сервіс був жахливим." (містить "жахливий", що є негативним словом).

3. Правило для нейтральної тональності:

- Якщо текст не містить явно позитивних або негативних слів, або містить рівну кількість позитивних та негативних слів, то тональність нейтральна.
- Наприклад, "Цей продукт виготовлено з пластика." (немає емоційно забарвлених слів).

4. Правило для врахування інтенсивності:

- Якщо текст містить слова, що підсилюють або послаблюють значення, це враховується при визначенні тональності.
- Наприклад, "Це просто чудово!" (містить "чудово" з підсилювачем "просто").

5. Правило для врахування заперечень:

- Якщо заперечення змінює значення слова, це враховується при визначенні тональності.
- Наприклад, "Цей фільм не був поганим." (містить заперечення "не", що змінює негативне "поганий" на позитивне).

Переваги:

1. Правила легко зрозуміти і пояснити. Користувачі можуть чітко бачити, як приймаються рішення.
2. Легко вносити зміни або налаштовувати правила для конкретних потреб або контекстів.
3. В порівнянні з методами машинного навчання, методи на основі правил можуть бути менш ресурсомісткими.

Недоліки:

1. Правила важко адаптувати до нових ситуацій або контекстів, які не були враховані при їх розробці.
2. Потребують регулярного оновлення та підтримки для збереження актуальності, особливо в умовах змін мови.
3. Методи на основі правил можуть мати труднощі з обробкою складних лінгвістичних конструкцій, таких як сарказм або іронія.
4. Збільшення кількості правил може ускладнити їхнє керування і підтримку.

Методи на основі правил є важливими інструментами для аналізу тональності, особливо в умовах, де потрібна прозорість та контрольованість. Однак, їхні обмеження роблять необхідним використання інших підходів, таких як машинне та глибоке навчання, для досягнення більш високої точності та гнучкості.

1.2.3 Методи машинного навчання

Наївний байєсівський класифікатор базується на застосуванні теореми Байєса з припущенням, що всі ознаки є незалежними[10]. Цей метод використовується для класифікації, де кожній події призначається ймовірність належності до певного класу.

Переваги:

- Дуже швидкий в навчанні та прогнозуванні.
- Потребує менше обчислювальних ресурсів.
- Добре працює навіть з наявністю шуму в даних.

Недоліки:

- Припущення про незалежність ознак рідко виконується в реальних даних, що може знизити точність.
- Може бути менш точним у порівнянні з іншими, більш складними моделями.

Метод опорних векторів (SVM) використовується для класифікації та регресії[10]. Він знаходить гіперплощину, яка максимально розділяє класи у багатовимірному просторі.

Переваги:

- Висока ефективність у випадках з чітко розділеними класами.
- Добре працює в просторах з високою розмірністю.
- Підтримує різні функції ядра (лінійні, поліноміальні, RBF).

Недоліки:

- Може бути повільним у навчанні та прогнозуванні для великих наборів даних.
- Вимагає ретельного налаштування параметрів для досягнення оптимальної продуктивності.

1.2.4 Методи глибокого навчання

Рекурентні нейронні мережі (RNN) є типом нейронних мереж, які мають петлі, що дозволяють зберігати інформацію про попередні стани[5,14]. Вони особливо ефективні для обробки послідовних даних, таких як текст або часоряди.

Переваги[15]:

- Добре підходять для аналізу тексту, аудіо та інших послідовних даних.
- Здатні зберігати контекст у послідовностях даних.

Недоліки[15]:

- Проблема зникаючих градієнтів у довгих послідовностях, що ускладнює навчання.
- Можуть мати проблеми зі збереженням довготривалої інформації.

LSTM є вдосконаленим типом RNN, який вирішує проблему зникаючих градієнтів[4]. Вони мають спеціальні блоки пам'яті, які дозволяють ефективно зберігати та передавати інформацію протягом тривалих послідовностей.

Переваги:

- Ефективні для задач, де важливо зберігати контекст протягом довгих послідовностей.
- Вирішують проблему зникаючих градієнтів у RNN.

Недоліки:

- Навчання LSTM є більш обчислювально складним та ресурсомістким.
- Більше параметрів для налаштування у порівнянні з традиційними RNN.

Трансформери (BERT, GPT) є новітнім підходом у глибокому навчанні для обробки природної мови[11]. Вони використовують механізм уваги, що дозволяє моделі ефективно зосереджуватись на різних частинах вхідних даних. BERT (Bidirectional Encoder Representations from Transformers) та GPT (Generative Pre-trained Transformer) є одними з найпопулярніших моделей на основі трансформерів[12].

- BERT - використовує двонаправлене навчання, що дозволяє моделі враховувати контекст з обох боків.
- GPT - фокусується на автопідказуванні та генерації тексту, використовуючи однонаправлене навчання.

Переваги:

- Моделі трансформерів можуть враховувати складні контекстуальні залежності.
- Показують високу точність у багатьох задачах обробки природної мови.
- Можуть бути використані для різних задач NLP, таких як класифікація, генерація тексту, переклад.

Недоліки:

- Навчання моделей трансформерів потребує великої кількості обчислювальних ресурсів.
- Для досягнення високої продуктивності потрібні великі обсяги навчальних даних.

1.3 Огляд нейронних мереж RNN, CNN, BERT

Рекурентні нейронні мережі (RNN) — це тип нейронних мереж, які мають петлі, що дозволяють зберігати інформацію про попередні стани[15]. Це робить їх особливо ефективними для обробки послідовних даних, таких як текст або часоряди.

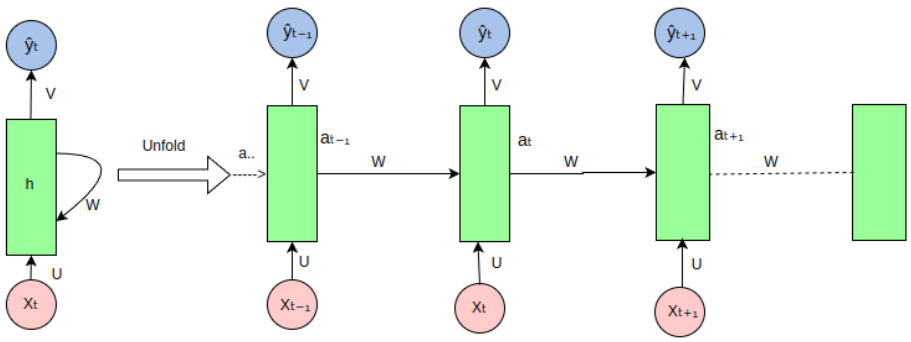


Рисунок 1.1 - Рекурентні нейронні мережі

Основні характеристики[15]:

- Моделі RNN здатні враховувати попередню інформацію у послідовності.
- Використання внутрішньої пам'яті для збереження інформації про попередні стани.

Переваги:

- Ідеально підходять для задач, де важливо враховувати порядок даних.
- Здатні зберігати контекст у послідовностях даних.

Недоліки:

- У довгих послідовностях градієнти можуть зникати або розростатися, що ускладнює навчання.
- Можуть мати проблеми зі збереженням довготривалої інформації.

Конволюційні нейронні мережі (CNN) — це тип нейронних мереж, розроблених для роботи з даними, які мають сіткову топологію (наприклад, зображення)[23,6]. Вони використовують конволюційні шари, які сканують вхідні дані та виявляють локальні патерни.

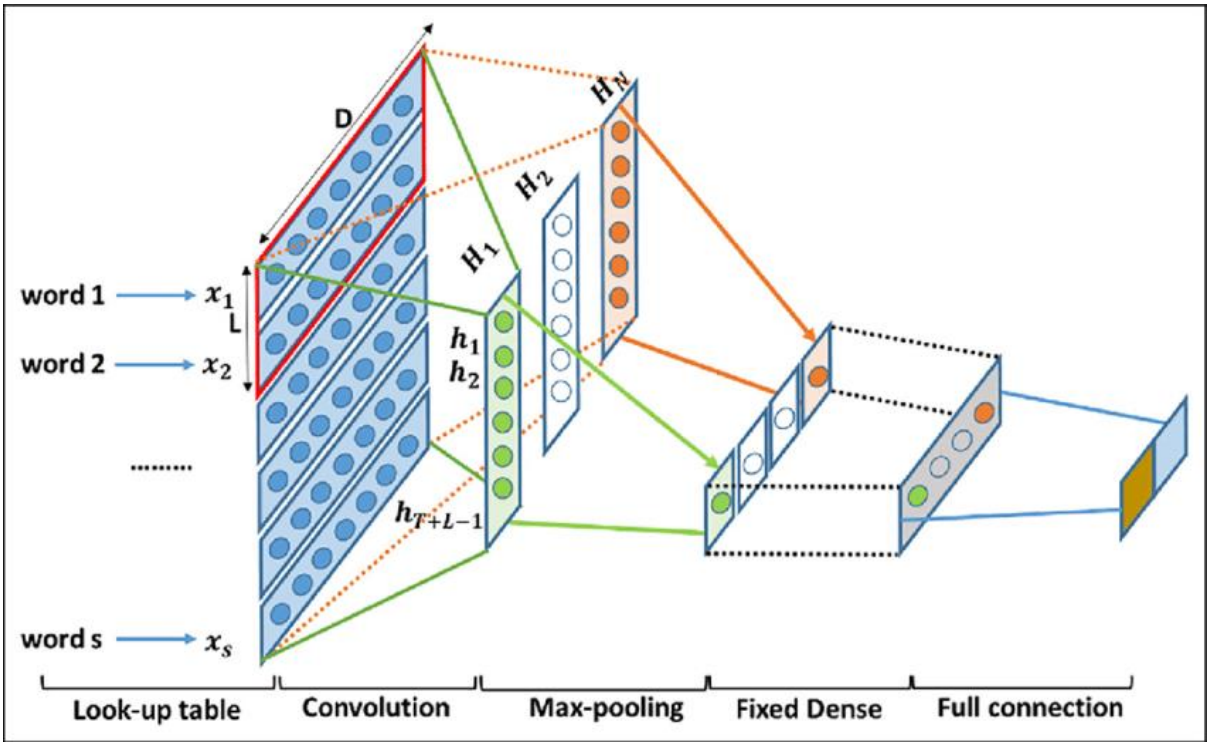


Рисунок 1.2 - Конволюційні нейронні мережі

Основні характеристики[6]:

- Використання фільтрів для виявлення локальних патернів у даних.
- Пулінг-шари зменшують розмірність даних, зберігаючи важливу інформацію.

Переваги:

- Добре підходять для задач розпізнавання образів та класифікації зображень.
- Можуть виявляти патерни незалежно від їх місцезнаходження у вхідних даних.

Недоліки:

- Менш ефективні для роботи з текстом або іншими послідовними даними.
- Потребують значних обчислювальних ресурсів, особливо для глибоких мереж.

Трансформери (BERT) є новітнім підходом у глибокому навчанні для обробки природної мови. BERT (Bidirectional Encoder Representations from Transformers) — це одна з найпопулярніших моделей трансформерів, яка використовує двонаправлене навчання для врахування контексту з обох боків[11].

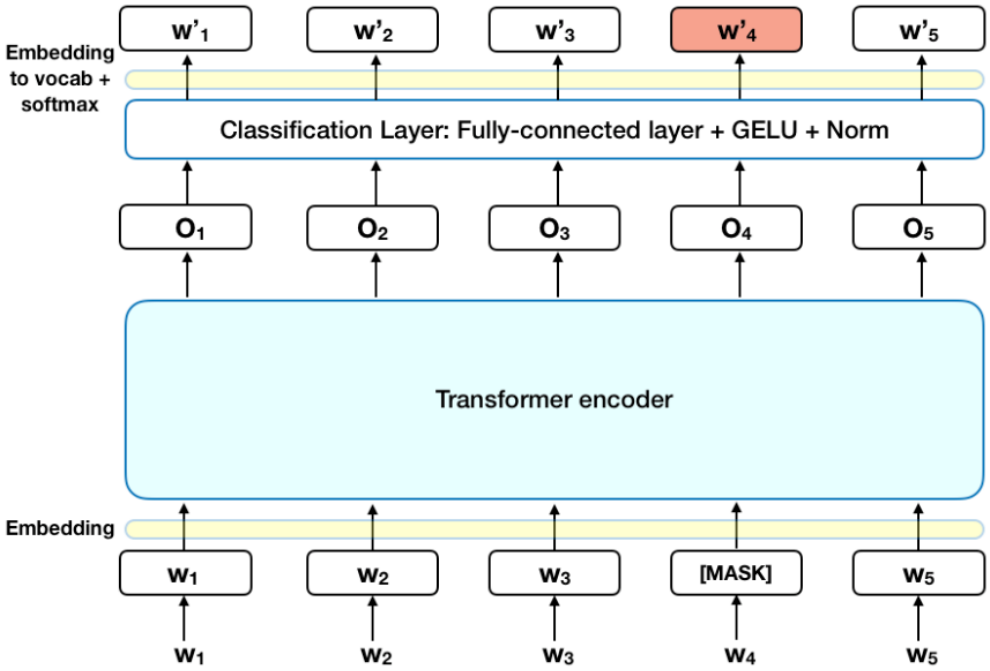


Рисунок 1.3 - Трансформери (BERT)

Основні характеристики[11,12,24]:

- Використовує механізм самоуваги для ефективного зосередження на різних частинах вхідних даних.
- Враховує контекст як зліва, так і справа від поточного слова.

Переваги:

- Показує високу точність у багатьох задачах обробки природної мови.
- Може бути використана для різних задач NLP, таких як класифікація, генерація тексту, переклад.

Недоліки:

- Навчання моделей трансформерів потребує великої кількості обчислювальних ресурсів.
- Для досягнення високої продуктивності потрібні великі обсяги навчальних даних.

Таблиця 1.4

Порівняння нейронних мереж: RNN, CNN, BERT

Характеристика	RNN	CNN	BERT
Тип даних	Послідовні дані (текст, часоряди)	Зображення, відео	Текстові дані (NLP)
Основна архітектура	Рекурентні шари	Конволюційні та пулінг-шари	Трансформер, механізм уваги
Збереження контексту	Так	Ні	Так (двонаправлене навчання)
Проблеми з навчанням	Зникаючі градієнти	Великі обчислювальні витрати	Великі обчислювальні витрати
Застосування	Обробка природної мови, часоряди	Розпізнавання образів, класифікація зображень	Класифікація тексту, генерація тексту, переклад

В таблиці 1.4 здійснено порівняння RNN, CNN і BERT за основними характеристиками, застосуваннями, перевагами та недоліками, допомагаючи зрозуміти, в яких ситуаціях кожна з цих нейронних мереж буде найбільш ефективною.

Провівши аналіз нейронних мереж для розробки було обрано BERT (Bidirectional Encoder Representations from Transformers), який є ефективним вибором для розробки алгоритмів машинного навчання для аналізу сентименту та емоцій в Instagram.

Висновки до розділу 1

У цьому розділі було проведено детальний огляд та аналіз методів машинного навчання і обробки природної мови (NLP). Розглянуто основні підходи та техніки, що використовуються в сучасній обробці текстової інформації. Було розглянуто основи керованого навчання, при якому алгоритми навчаються на мічених даних. Детально описано приклади алгоритмів, таких як логістична регресія, дерева рішень, метод опорних векторів (SVM) та нейронні мережі, а також їхнє застосування в задачах класифікації та регресії. Вивчено методи некерованого навчання, де моделі працюють з неміченими даними. Розглянуто алгоритми кластеризації (k-means, ієрархічна кластеризація) та зменшення розмірності (PCA, t-SNE), а також їхнє застосування в аналізі даних. Описано методи напівкерованого навчання, які використовують як мічені, так і немічені дані. Було розглянуто їхнє застосування в умовах, коли мічених даних недостатньо для ефективного навчання моделі. Розглянуто процес токенізації, що включає розбиття тексту на окремі слова або токени. Обговорено різні підходи до токенізації та їх важливість у підготовці текстових даних для подальшої обробки.

Проаналізовано концепцію стоп-слів — слів, які часто видаляються з текстів перед обробкою через їхню високу частотність та низьку інформативність. Наведено приклади стоп-слів та обговорено їхнє значення в NLP. Вивчено методи векторизації тексту, які перетворюють текстові дані у числові вектори для подальшої обробки моделями машинного навчання. Описано різні підходи до векторизації, включаючи Bag of Words, TF-IDF та word embeddings. Було досліджено різні методи класифікації тональності тексту, що дозволяють визначити емоційне забарвлення повідомлення.

Описано використання методів машинного навчання для класифікації тональності, зокрема наївного байєсівського класифікатора, методу опорних векторів (SVM) та логістичної регресії. Обговорено їхні переваги та недоліки у порівнянні з іншими підходами. Розглянуто застосування методів глибокого навчання для класифікації тональності, включаючи рекурентні нейронні мережі

(RNN), довготривалу короткочасну пам'ять (LSTM) та трансформери (BERT, GPT). Проаналізовано їхню ефективність та виклики.

2. АРХІТЕКТУРА СИСТЕМИ ДЛЯ АНАЛІЗУ ТОНАЛЬНОСТІ КОМЕНТАРІВ

2.1 Визначення вимог до системи

2.1.1 Функціональні вимоги

Функціональні вимоги описують, що саме система повинна робити. Вони визначають конкретні функції, можливості та задачі, які система повинна виконувати для задоволення потреб користувачів[26]. Функціональні вимоги охоплюють всі аспекти поведінки системи, включаючи введення, обробку, виведення даних та взаємодію з іншими системами.

Отже, система повинна забезпечувати наступні функціональні вимоги:

1. Збір даних

- Система повинна автоматично збирати коментарі з Instagram, використовуючи API платформи або інші методи парсингу.
- Система повинна мати можливість фільтрувати спам та неінформативні коментарі для підвищення якості аналізу.

2. Попередня обробка тексту

- Система повинна розбивати текст коментарів на окремі слова або фрази.
- Система повинна видаляти поширені, неінформативні слова.
- Система повинна приводити слова до їх базової форми для покращення аналізу.

3. Векторизація тексту

- Система повинна перетворювати текст коментарів у числові вектори, використовуючи моделі представлення слів, такі як BERT.

4. Аналіз сентименту та емоцій

- Система повинна визначати, чи є коментар позитивним, негативним або нейтральним.
- Система повинна розпізнавати різні емоції в коментарях (радість, сум, злість тощо).

5. Візуалізація результатів

- Система повинна мати користувацький інтерфейс для відображення результатів аналізу у вигляді графіків, діаграм тощо.
- Система повинна генерувати звіти та надавати аналітичну інформацію про результати аналізу.

2.1.2 Нефункціональні вимоги

Нефункціональні вимоги описують, як система повинна виконувати свої функції. Вони визначають атрибути якості, обмеження та інші характеристики, які впливають на продуктивність, надійність, безпеку, зручність використання та інші аспекти роботи системи[26]. Нефункціональні вимоги не пов'язані з конкретними функціями, але визначають загальні критерії, які система повинна задовольняти.

Отже, система повинна забезпечувати наступні нефункціональні вимоги:

1. Продуктивність

- Система повинна мати високу швидкість обробки коментарів, забезпечуючи аналіз великого обсягу даних у реальному часі.
- Система повинна бути здатна обробляти збільшені обсяги даних без значного зниження продуктивності.

2. Надійність

- Система повинна забезпечувати високу доступність та безперебійність роботи.
- Система повинна мати можливість швидкого відновлення після збоїв.

3. Безпека

- Система повинна забезпечувати захист даних від несанкціонованого доступу та витоків інформації.

4. Зручність використання

- Система повинна мати зручний та інтуїтивно зрозумілий інтерфейс для користувачів.

5. Сумісність

- Система повинна легко інтегруватися з іншими інформаційними системами та платформами.
- Система повинна підтримувати різні формати даних для імпорту та експорту результатів аналізу.

2.2 Вибір інструментів та технологій

Для розробки алгоритмів машинного навчання для аналізу сентименту та емоцій в коментарях Instagram було обрано наступні інструменти та технології: Python і BERT. Нижче наведено детальний опис кожного з них.

Python — це високорівнева мова програмування загального призначення, відома своєю простотою та читабельністю. Вона має широкий спектр бібліотек та інструментів, які роблять її ідеальним вибором для задач машинного навчання та обробки природної мови (NLP).

Переваги використання Python:

1. Python має простий синтаксис, що полегшує написання та читання коду. Це дозволяє швидко розробляти та тестувати алгоритми.
2. Python має великий набір бібліотек для машинного навчання (TensorFlow, PyTorch, scikit-learn), обробки тексту (NLTK, spaCy), роботи з даними (pandas, NumPy) та візуалізації (matplotlib, seaborn).
3. Python працює на різних операційних системах, включаючи Windows, macOS та Linux, що дозволяє розробникам використовувати ту платформу, яка їм зручна.

Основні бібліотеки для задачі:

- Transformers - бібліотека від Hugging Face, яка надає простий інтерфейс для роботи з моделями BERT та іншими трансформерами.
- pandas - бібліотека для обробки та аналізу даних, що дозволяє легко маніпулювати таблицями та структурами даних.
- scikit-learn - бібліотека для машинного навчання, яка включає інструменти для попередньої обробки, моделювання та оцінки моделей.

- NumPy - бібліотека для роботи з багатовимірними масивами та матрицями, яка є основою для багатьох інших наукових бібліотек Python.

BERT (Bidirectional Encoder Representations from Transformers) — це модель трансформера для обробки природної мови, розроблена компанією Google. BERT використовує двонаправлене навчання для врахування контексту з обох боків слова, що дозволяє моделі краще розуміти значення та взаємозв'язок слів у реченні.

Переваги використання BERT:

1. На відміну від традиційних моделей, які обробляють текст послідовно, BERT враховує контекст з обох боків слова, що підвищує точність розуміння тексту.
2. BERT показує високу точність у багатьох задачах обробки природної мови, таких як аналіз тональності, класифікація тексту, відповіді на питання тощо.
3. BERT можна донавчити на специфічних даних для покращення продуктивності в конкретних задачах або доменах.

Використання BERT у задачі аналізу сентименту та емоцій:

1. Використання спеціального токенізатора BERT для перетворення тексту коментарів у формат, який може бути використаний моделлю.
2. Використання попередньо навченої моделі BERT та її донавчання на специфічних даних з Instagram для аналізу сентименту та емоцій.
3. Використання донавченої моделі BERT для класифікації коментарів як позитивних, негативних, нейтральних або для виявлення специфічних емоцій.

2.3 Архітектура програмного забезпечення

2.3.1 Діаграма використання

Діаграма використання (Use Case Diagram) (Рис. 2.1) є одним з типів діаграм UML (Unified Modeling Language), що використовуються для моделювання функціональних вимог системи [27]. Вона показує взаємодію між користувачами (акторами) та системою через набір випадків використання (use cases). Основна мета такої діаграми — візуалізація того, як різні користувачі взаємодіють із системою та які функції вони використовують.

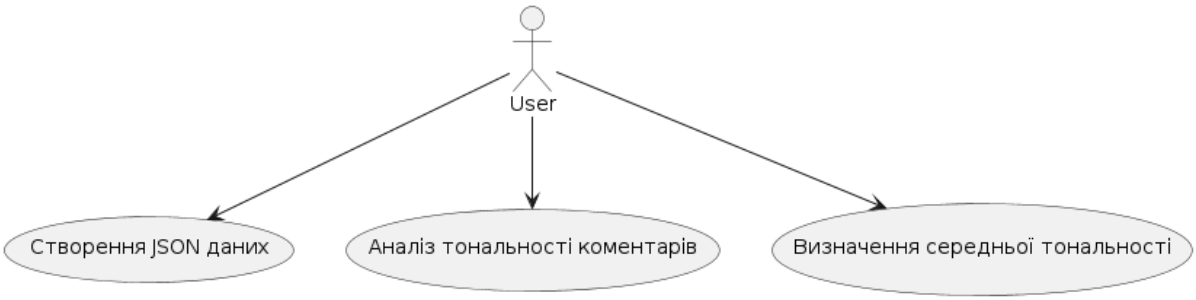


Рисунок 2.1 - Діаграма використання

На даній діаграмі використання показано взаємодію актора User (користувач) із системою, яка надає три основні функції:

1. Створення JSON даних (UC1) - цей випадок використання показує, що користувач може створювати JSON дані. Це стосується створення вхідних даних для подальшого аналізу. Користувач ініціює процес створення JSON даних.

2. Аналіз тональності коментарів (UC2) - цей випадок використання передбачає, що користувач може запускати процес аналізу тональності коментарів. Це включає класифікацію коментарів як позитивних, негативних або нейтральних. Користувач ініціює аналіз тональності коментарів.

3. Визначення середньої тональності (UC3) - цей випадок використання дозволяє користувачу визначити середню тональність коментарів. Це означає обчислення середньої оцінки тональності на основі результатів аналізу. Користувач ініціює процес визначення середньої тональності коментарів.

2.3.2 Діаграма розгортання

Діаграма розгортання (Deployment Diagram) є типом діаграми UML, яка показує фізичну архітектуру системи. Вона відображає, як апаратні та програмні компоненти системи розгорнуті на фізичних машинах (вузлах)[27]. Ця діаграма важлива для розуміння того, як система буде функціонувати в реальному середовищі та як компоненти взаємодіють один з одним на фізичному рівні.

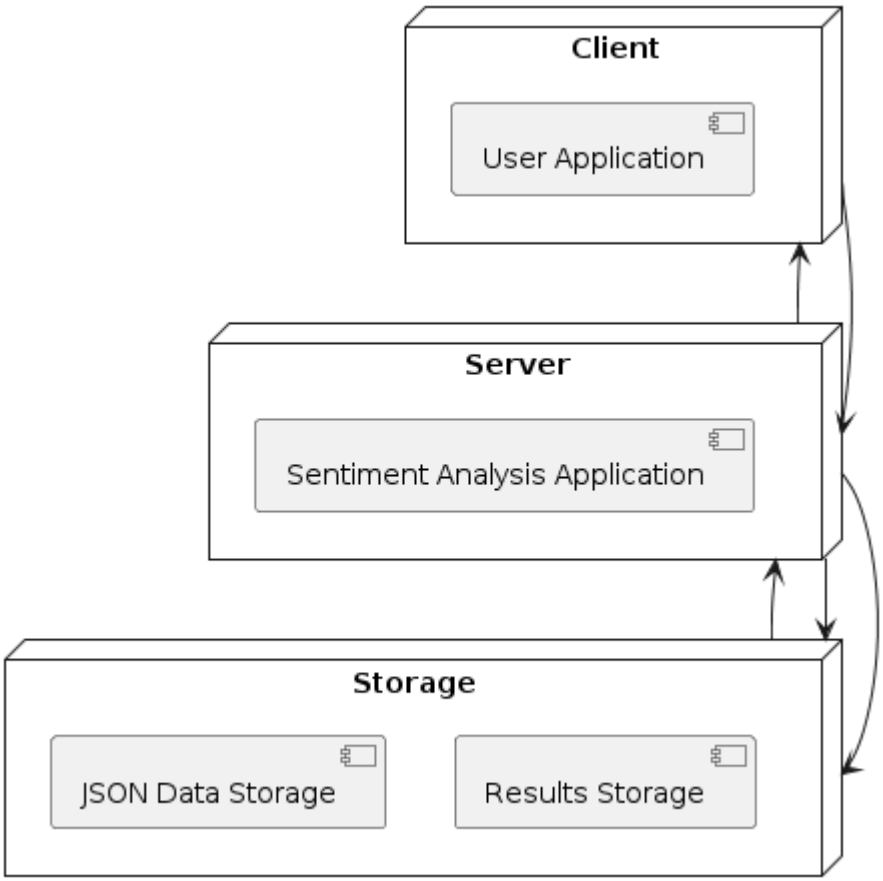


Рисунок 2.2 - Діаграма розгортання

Компоненти системи

Клієнт

- Виконує програму для створення JSON даних.
- Відправляє запит на сервер для аналізу тональності.
- Отримує результати аналізу.

Сервер для аналізу

- Виконує обчислення тональності коментарів.
- Зберігає результати аналізу.

Зберігання даних

- Зберігає початкові дані (пости та коментарі) у форматі JSON.
- Зберігає результати аналізу у форматі JSON.

Пояснення

Client - вузол, де користувач запускає програму для створення JSON даних та відправки запитів на сервер для аналізу.

Server - вузол, де виконується програма для аналізу тональності коментарів. Сервер отримує дані від клієнта, обробляє їх і зберігає результати.

Storage - вузол для зберігання даних, включаючи початкові дані JSON (пости та коментарі) та результати аналізу.

2.3.3 Діаграма послідовності

Діаграма послідовності (Sequence Diagram) є типом діаграми UML, яка показує взаємодію між об'єктами в системі у хронологічному порядку. Вона відображає послідовність повідомлень, що передаються між об'єктами для виконання певної функції або процесу[27]. Ця діаграма допомагає зрозуміти динаміку системи, тобто як елементи системи взаємодіють один з одним протягом часу.

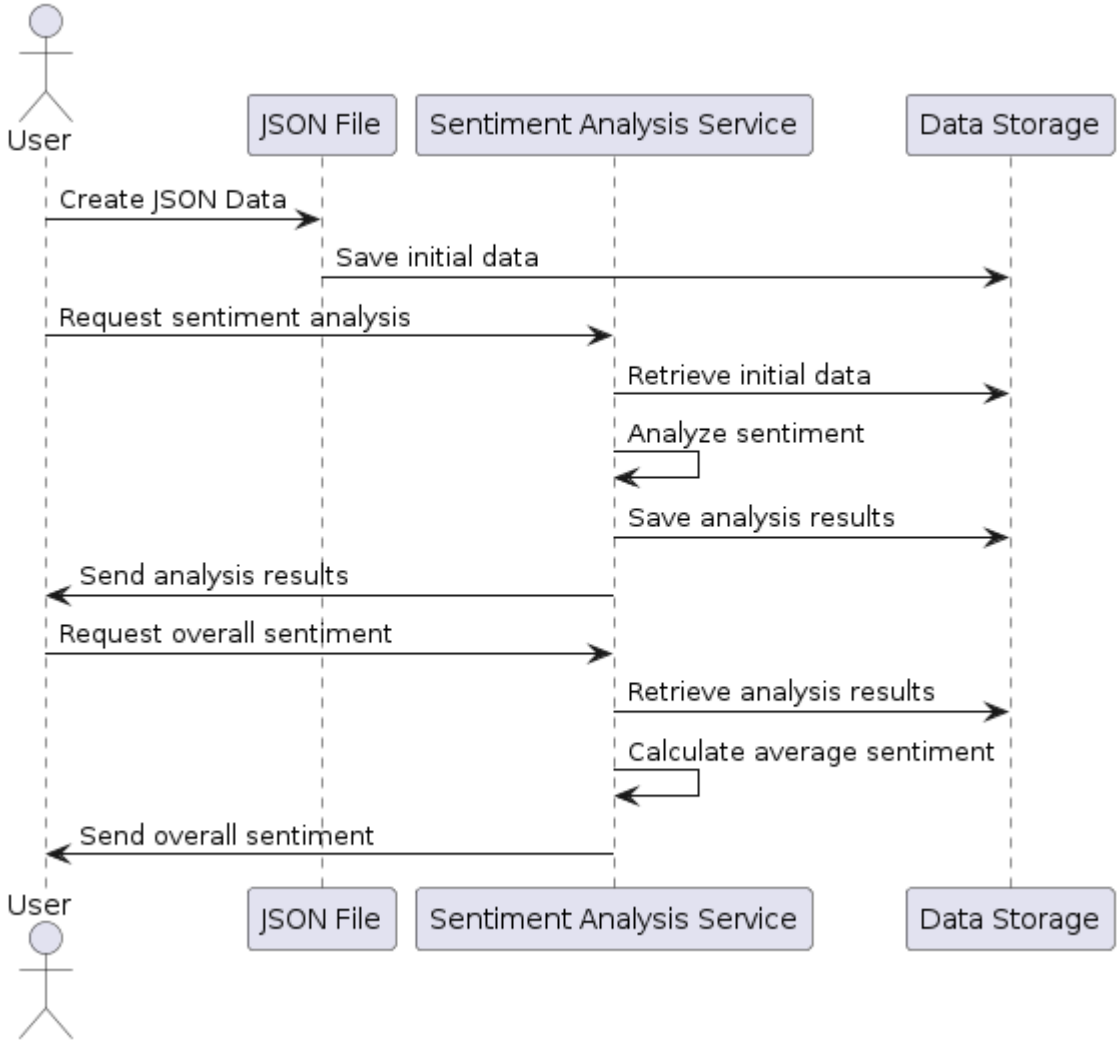


Рисунок 2.3 - Діаграма послідовності

Актори та учасники:

- User - Користувач, який ініціює процеси.
- JSON File - об'єкт, який створює JSON дані.
- Sentiment Analysis Service - сервіс для аналізу тональності коментарів.
- Data Storage - сховище даних, де зберігаються початкові дані та результати аналізу.

Послідовність взаємодій:

1. Користувач створює JSON дані (Create JSON Data).
2. JSON дані зберігаються у сховищі (Save initial data).
3. Користувач запитує аналіз тональності (Request sentiment analysis).
4. Сервіс для аналізу тональності отримує початкові дані зі сховища (Retrieve initial data).

5. Сервіс аналізує тональність коментарів (Analyze sentiment).
6. Результати аналізу зберігаються у сховищі (Save analysis results).
7. Сервіс відправляє результати аналізу користувачу (Send analysis results).
8. Користувач запитує загальну емоційну забарвленість (Request overall sentiment).
9. Сервіс отримує результати аналізу зі сховища (Retrieve analysis results).
10. Сервіс обчислює середню тональність (Calculate average sentiment).
11. Сервіс відправляє загальну емоційну забарвленість користувачу (Send overall sentiment).

2.3.3 Діаграма компонентів

Діаграма компонентів (Component Diagram) є типом діаграми UML, яка показує організацію та залежності між програмними компонентами в системі[27]. Вона відображає, як компоненти програмного забезпечення взаємодіють між собою і як вони об'єднані в більші структури. Ця діаграма корисна для моделювання високорівневої архітектури системи та розуміння залежностей між різними частинами програмного забезпечення.

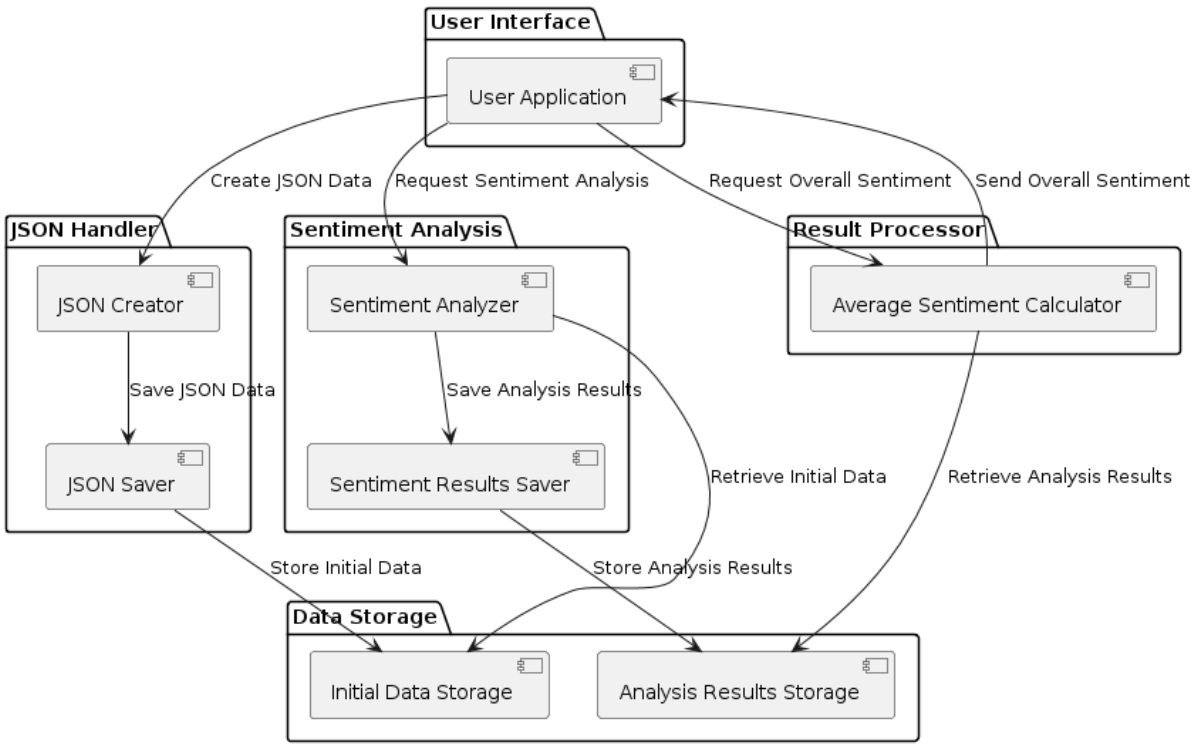


Рисунок 2.4 - Діаграма компонентів

Отже, є 5 компонент:

1. User Interface (UI) - компонента, яка взаємодіє з користувачем для створення JSON даних, відправки запитів на аналіз та отримання результатів.
2. JSON Handler:
   - JSON Creator - компонента, яка створює JSON дані.
   - JSON Saver - компонента, яка зберігає JSON дані.
3. Sentiment Analysis:
   - Sentiment Analyzer - компонента, яка виконує аналіз тональності коментарів.
   - Sentiment Results Saver - компонента, яка зберігає результати аналізу.
4. Data Storage:
   - Initial Data Storage - сховище для початкових JSON даних.
   - Analysis Results Storage - сховище для результатів аналізу тональності.
5. Result Processor - компонента, яка обчислює середню емоційну забарвленість на основі результатів аналізу.

Висновки до розділу 2

У цьому розділі було детально розглянуто архітектуру системи для аналізу тональності коментарів. Було визначено основні функціональні та нефункціональні вимоги до системи аналізу тональності коментарів, вибрано відповідні інструменти та технології для реалізації цієї системи, а також розроблено архітектуру програмного забезпечення з використанням різних типів діаграм UML.

Показано взаємодію користувача із системою через три основні функції: створення JSON даних, аналіз тональності коментарів та визначення середньої тональності. Відображено фізичну архітектуру системи, включаючи клієнтський вузол, сервер для аналізу та сховище даних. Відображено послідовність взаємодій між користувачем, JSON файлом, сервісом для аналізу тональності та сховищем даних. Показано організацію та залежності між програмними компонентами системи.

3. РОЗРОБКА ТА ТЕСТУВАННЯ АЛГОРИТМІВ

3.1 Архітектура нейромережі

Архітектура нейромережі, яку ми використовуємо в даному випадку, базується на моделі RoBERTa, спеціально натренованій для аналізу тональності твітів (twitter-roberta-base-sentiment-latest). Ця модель є варіантом архітектури BERT (Bidirectional Encoder Representations from Transformers), яка використовує трансформерну архітектуру для обробки текстових даних.

1. Токенізатор (Tokenizer)

- AutoTokenizer: токенізатор відповідає за перетворення вхідного тексту у формат, який модель може обробити. Він розбиває текст на токени та перетворює їх у числові індекси, додаючи спеціальні токени для початку і кінця послідовності.
- Токенізація: процес включає нормалізацію тексту, розбиття на слова, додавання спеціальних токенів ([CLS], [SEP]) і перетворення в словникові індекси.

2. Модель RoBERTa (AutoModelForSequenceClassification)

- Трансформери: RoBERTa використовує багатошарову трансформерну архітектуру, яка включає енкодери, що дозволяють моделі враховувати контекст кожного слова в тексті двонаправлено (зліва направо і справа наліво).
- Класифікація: вихідний шар моделі (голова класифікації) відповідає за прогнозування тональності тексту. Вона складається з одного або декількох шарів нейронних мереж з функцією активації softmax для отримання ймовірностей для кожного класу (негативний, нейтральний, позитивний).

3.2 Опис методів

Для розробки системи було запрограмовано 3 програмних файли:

1. analyze.py

Завдання: Класифікація тональності коментарів.

Основні функції:

- Завантаження локальної моделі та токенізатора.
- Передбачення тональності тексту (коментарів).
- Обробка коментарів і збереження результатів у JSON форматі.

2. comm.py

Завдання: Підготовка даних для аналізу.

Основні функції:

- Створення структури даних постів і коментарів.
- Збереження підготовлених даних у JSON файл.

3. avg.py

Завдання: Аналіз і візуалізація середньої тональності коментарів.

Основні функції:

- Завантаження класифікованих коментарів.
- Обчислення середніх ймовірностей для кожного класу тональності.
- Визначення загальної емоційної забарвленості поста.
- Створення кругової діаграми для візуалізації результатів.

Розглянемо кожен із файлів детально, опишемо методи.

Файл analyze.py:

1. load_local_model()
   - Опис: завантажує локальний токенізатор і модель для класифікації тексту.
   - Параметри: немає.
   - Повертає: повертає об'єкти токенізатора та моделі.
2. predict_sentiment(tokenizer, model, text)
   - Опис: передбачає тональність (емоційну забарвленість) тексту.
   - Параметри:
   - tokenizer (AutoTokenizer): токенізатор для підготовки тексту.
   - model (AutoModelForSequenceClassification): модель для класифікації.
   - text (str): текст для аналізу.

   Повертає: Кортеж, що містить:
   - sentiment (int): передбачений клас тональності (0 - негативний, 1 - нейтральний, 2 - позитивний).
   - probabilities (Tensor): ймовірності для кожного класу.

Процес аналізу

1. Завантаження даних: завантажує коментарі з файлу posts_and_comments.json.
2. Класифікація коментарів: для кожного коментаря викликає функцію predict_sentiment, результати зберігаються у списку results.
3. Збереження результатів: записує результати класифікації в файл classified_comments.json.

Файл comm.py:

1. Підготовка даних
   - Опис: створює список постів з коментарями для подальшого аналізу.
   - Дані посту і коментарів: зберігає інформацію про пост і коментарі у форматі JSON.
   - Збереження даних: записує дані у файл posts_and_comments.json.

Файл avg.py

1. Завантаження класифікованих коментарів
   - Опис: завантажує результати класифікації з файлу classified_comments.json.
2. Обчислення середніх ймовірностей
   - Опис: підраховує суму ймовірностей для кожного класу тональності та обчислює середні значення.
   - Ініціалізація змінних: ініціалізує змінні total_probabilities та count.
   - Підрахунок ймовірностей: для кожного коментаря додає ймовірності до total_probabilities.
3. Визначення загальної емоційної забарвленості
   - Опис: визначає загальну тональність поста на основі середніх ймовірностей.
   - Параметри: використовує sentiment_mapping для відображення класів тональності.
4. Візуалізація результатів
   - Опис: створює кругову діаграму, що відображає співвідношення настроїв коментарів.
   - Використовує бібліотеки: numpy, matplotlib.pyplot.

- Результат: виводить кругову діаграму з підписами та процентами для кожного класу тональності.

3.3 Тестування системи

Перейдемо до тестування системи. Для початку створимо файл JSON з коментарями до посту. Для цього запустимо файл comm.py. Результат відображено на рисунку 3.1.


```
[
    {
        "post_id": "1",
        "post_text": "Kevin De Bruyne is one of the greatest playmakers in PL history  112 assists (2nd all-time) 10+ assists in six different seasons\n\u25aa\ufe0f 10 assists in just 17 games this year",
        "comments": [
            "greatest player to ever grace the premier league.",
            "I don\u2019t know why Raheem is included in this picture, Kevin could have been 60+ assist up of Raheem converted all those chances. The guy missed like he was competing with Nunez",
            "KVD > > > Magician of football world",
            "It should be Greatest not one of the Greatest.",
            "Top 3 greatest midfielders of all time",
            "He is one of the greatest ever in footballs history ",
            "But people will argue with me that Ozil is clear smh",
            "You will always stand as a legend to me",
            "GOAT premier league midfielder and it's not even close",
            "HONESTLY SHE IS SO GOOD ",
            "Without him halaand is nothing"
        ]
    }
]
```


Рисунок 3.1 - Створення JSON файлу

Наступним кроком є аналіз тональності всіх коментарів, для цього запустимо файл analize.py. Дані зберігаються у json файлі. Результат відображено на рисунку 3.2.


```
{
    "post_id": "1",
    "comment": "KVD > > > Magician of football world",
    "sentiment": "positive",
    "probabilities": [
        [
            0.013346608728170395,
            0.4164392948150635,
            0.5702140927314758
        ]
    ]
},
{
    "post_id": "1",
    "comment": "It should be Greatest not one of the Greatest.",
    "sentiment": "neutral",
    "probabilities": [
        [
            0.25092369318008423,
            0.6003116369247437,
            0.1487647145986557
        ]
    ]
},
{
    "post_id": "1",
    "comment": "Top 3 greatest midfielders of all time",
    "sentiment": "positive",
    "probabilities": [
        [
            0.012974844314157963,
            0.1434151530265808,
            0.8436099886894226
        ]
    ]
},
```


Рисунок 3.2 - Аналіз тональності коментарів

Наступним кроком є визначення загальної тональності коментарів під постом, для цього запустимо файл avg.py. Отримуємо діаграму з розподіленням тональності та загальний результат. Результат відображено на рисунку 3.3 та 3.4.

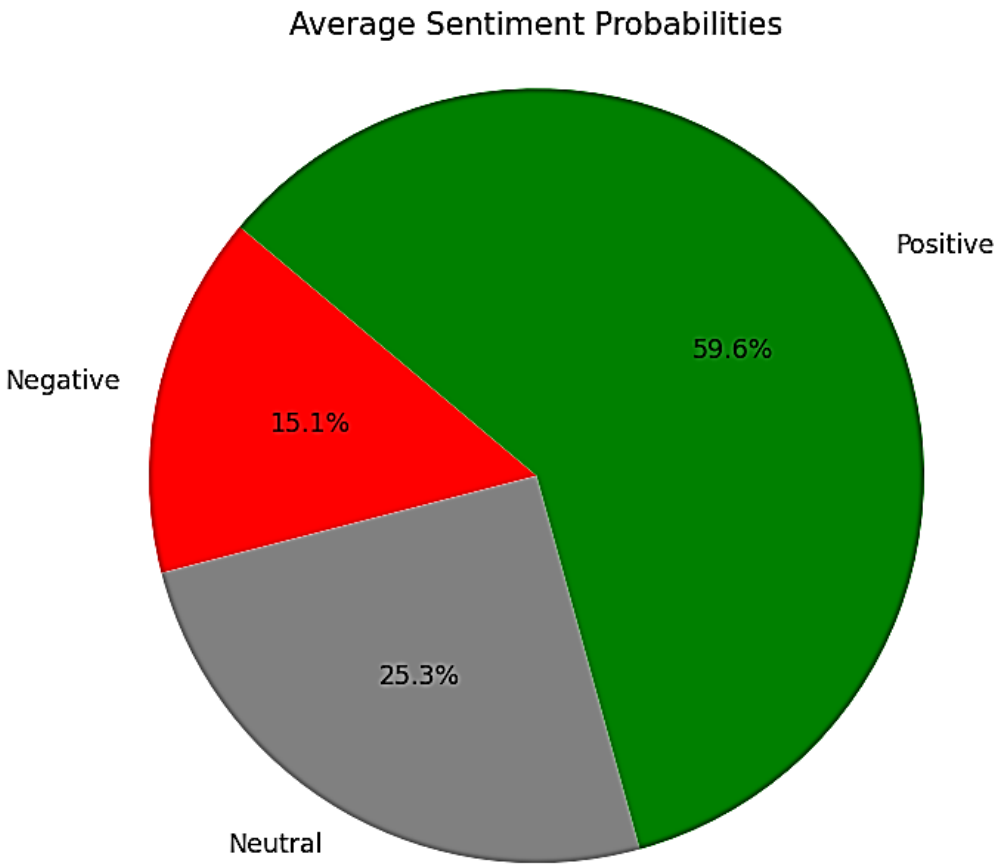


Рисунок 3.3 - Розподілення тональності коментарів під постом

На рисунку 3.3 відображена кругова діаграма, що відображає співвідношення середніх ймовірностей тональності коментарів. Діаграма містить три сектори, які представляють різні класи тональності:

1. Positive (Позитивна)
    - Ймовірність: 59.6%
    - Колір: Зелений
    - Опис: Найбільший сектор, що свідчить про те, що більшість коментарів мають позитивну тональність.
2. Neutral (Нейтральна)
    - Ймовірність: 25.3%
    - Колір: Сірий

- Опис: Другий за розміром сектор, що показує, що чверть коментарів мають нейтральну тональність.

3. Negative (Негативна)
    - Ймовірність: 15.1%
    - Колір: Червоний
    - Опис: Найменший сектор, який свідчить про те, що менша частина коментарів має негативну тональність.

Згідно з даними, більшість коментарів до посту мають позитивну емоційну забарвленість (59.6%), значна частина є нейтральними (25.3%), і меншість є негативними (15.1%). Це свідчить про загально позитивне сприйняття посту серед користувачів.

```
Average Probabilities: [0.15148008 0.25256835 0.59595158]
Overall Sentiment: positive
```

Рисунок 3.4 - Загальний результат

Average Probabilities: [0.15148008, 0.25256835, 0.59595158]

Це середні ймовірності для кожного з класів (негативний, нейтральний, позитивний) на основі аналізу тональності коментарів. Давайте розберемося, що означають ці значення:

1. 0.15148008 (Негативний): середня ймовірність того, що коментар є негативним, становить приблизно 15.15%.
2. 0.25256835 (Нейтральний): середня ймовірність того, що коментар є нейтральним, становить приблизно 25.26%.
3. 0.59595158 (Позитивний): середня ймовірність того, що коментар є позитивним, становить приблизно 59.60%.

Overall Sentiment: positive - це загальна емоційна забарвленість поста на основі аналізу всіх коментарів. У цьому випадку загальна тональність визначена як позитивна, оскільки середня ймовірність для позитивного класу є найвищою серед усіх класів (59.60%).

Пояснення результату

1. Негативний (15.15%): низька середня ймовірність для негативного класу означає, що коментарі до цього посту рідко містять негативні емоції.
2. Нейтральний (25.26%): середня ймовірність для нейтрального класу показує, що частина коментарів є нейтральними, але цей клас не домінує.
3. Позитивний (59.60%): висока середня ймовірність для позитивного класу вказує на те, що більшість коментарів до посту є позитивними, що й визначило загальну емоційну забарвленість як позитивну.

На основі аналізу тональності коментарів можна зробити висновок, що пост отримав переважно позитивні відгуки від користувачів. Висока середня ймовірність для позитивного класу (59.60%) свідчить про те, що більшість коментарів містять позитивні емоції. Це може бути корисною інформацією для розуміння реакції аудиторії на пост та подальшого планування контенту.

3.4 Метрики оцінювання роботи системи

Для оцінювання роботи системи класифікації тональності коментарів можна використовувати кілька метрик. Ці метрики допоможуть визначити, наскільки точно модель класифікує коментарі в різні категорії (негативний, нейтральний, позитивний).

Основні метрики, які використовуються:

1. Accuracy (Точність) - відсоток правильних передбачень від загальної кількості передбачень.
2. Precision (Точність для класу) - відсоток правильних позитивних передбачень для кожного класу.
3. Recall (Повнота) - відсоток правильних передбачень серед всіх фактичних позитивних випадків для кожного класу.
4. F1-Score (F1-метрика) - гармонійне середнє між точністю та повнотою.

Ми перевіряємо, чи є коментар у словнику true_sentiments. Якщо є, додаємо відповідні значення до списків true_labels та predicted_labels.

Обчислюємо метрики лише тоді, коли обидва списки true_labels і predicted_labels не пусті. Якщо немає істинних міток, виводимо повідомлення, що метрики не можуть бути обчислені. Цей підхід дозволяє обчислювати метрики, якщо є істинні мітки, і уникати помилок, коли істинні мітки відсутні.

Таблиця 3.1

Результати метрик для коментарів

	Accuracy	Precision	Recall	F1
Текст 1	90%	76%	79%	72%
Текст 2	91%	74%	71%	76%
Текст 3	89%	69%	74%	73%
Текст 4	93%	75%	80%	75%

Повні тексти розміщено у додатках.

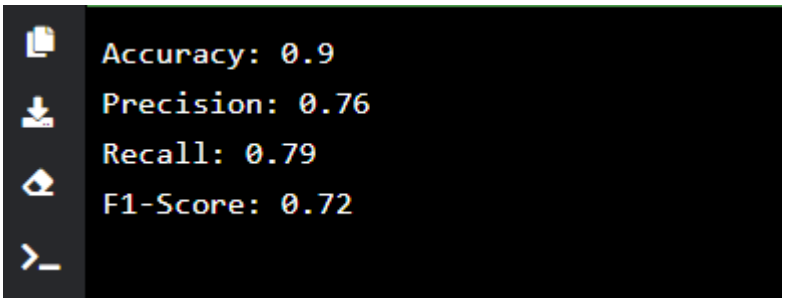


Рисунок 3.5 - Метрики

Accuracy (Точність): 0.90 - це означає, що 90% усіх передбачень моделі були правильними. Це досить високий показник, що вказує на загальну ефективність моделі.

Precision (Точність): 0.76 - це означає, що 76% передбачених позитивних результатів були правильними. Тобто, з усіх випадків, коли модель передбачила позитивний результат, 76% дійсно були позитивними. Це показує, що модель має певну кількість хибнопозитивних результатів.

Recall (Повнота): 0.79 - це означає, що модель правильно ідентифікувала 79% всіх фактичних позитивних випадків. Іншими словами, з усіх фактичних позитивних випадків модель правильно ідентифікувала 79%. Результат показує, що модель має певну кількість хибнонегативних результатів.

F1-Score є гармонійним середнім між Precision і Recall. Значення 0.72 вказує на те, що модель має збалансовану ефективність між точністю і повнотою, хоча і не ідеальну. Це важливо в ситуаціях, коли потрібно балансувати між двома метриками, і коли високий Precision або Recall поодинці не є достатнім.

Висновки до розділу 3

У даному розділі було розроблено архітектуру нейромережі для класифікації тональності коментарів на основі моделі RoBERTa, спеціально натренованої для аналізу тональності твітів. Модель використовує трансформерну архітектуру, яка забезпечує двонаправлену обробку текстових даних, дозволяючи враховувати контекст кожного слова в тексті.

Результати тестування показали, що більшість коментарів до посту мають позитивну тональність (59.6%), значна частина є нейтральними (25.3%) і меншість є негативними (15.1%). Це свідчить про загально позитивне сприйняття поста серед користувачів.

4. ІНСТРУКЦІЯ КОРИСТУВАЧА

Дана програма дозволяє аналізувати коментарі до постів, визначаючи їхню емоційну забарвленість (тональність) за допомогою нейронної мережі. Процес складається з трьох основних етапів: формування файлу з коментарями, визначення тональності за допомогою нейронної мережі та розрахунок середньої тональності для всіх коментарів.

Етап 1. Формування JSON файлу з коментарями

Крок 1. Створення файлу generate_comments_json.py

1. Відкрийте ваш текстовий редактор або IDE.
2. Створіть новий файл з назвою generate_comments_json.py.
3. Додайте код до файлу.

Крок 2. Запуск файлу generate_comments_json.py

1. Відкрийте термінал або командний рядок.
2. Перейдіть до директорії, де зберігається ваш файл generate_comments_json.py.
3. Запустіть файл.
4. Після виконання цієї команди у вашій директорії з'явиться файл posts_and_comments.json, який міститиме дані посту та коментарів.

Етап 2. Визначення тональності за допомогою нейронної мережі

Крок 3. Створення файлу classify_comments.py

1. Створіть новий файл з назвою classify_comments.py.
2. Додайте код до файлу.

Крок 4. Запуск файлу classify_comments.py

1. Переконайтеся, що у вас є локально завантажені модель та токенізатор local_twitter_roberta.
2. Відкрийте термінал або командний рядок.
3. Перейдіть до директорії, де зберігається ваш файл classify_comments.py.
4. Запустіть файл.
5. Після виконання цієї команди у вашій директорії з'явиться файл classified_comments.json, який міститиме результати класифікації коментарів за тональністю.

Етап 3. Розрахунок середньої тональності для всіх коментарів

Крок 5. Створення файлу calculate_average_sentiment.py

1. Створіть новий файл з назвою calculate_average_sentiment.py.
2. Додайте код до файлу.

Крок 6. Запуск файлу calculate_average_sentiment.py

1. Відкрийте термінал або командний рядок.
2. Перейдіть до директорії, де зберігається ваш файл calculate_average_sentiment.py.
3. Запустіть файл.
4. Після виконання цієї команди програма розрахує середню тональність для всіх коментарів і виведе кругову діаграму співвідношення настроїв.

Виконуючи ці кроки, ви зможете отримати загальну емоційну забарвленість коментарів до постів.

ВИСНОВКИ

У рамках дипломної роботи було здійснено всебічне дослідження, аналіз та розробку системи для аналізу тональності коментарів з використанням методів машинного навчання та обробки природної мови (NLP). Вивчено різні підходи до машинного навчання, включаючи кероване, некероване, напівкероване навчання та навчання з підкріпленням.

Розглянуто основні методи обробки тексту, такі як токенізація, видалення стоп-слів, лемматизація та стемінг, а також векторизація тексту. Проаналізовано лексичні методи, методи на основі правил, методи машинного навчання та методи глибокого навчання для класифікації тональності тексту.

Вивчено особливості та застосування різних типів нейронних мереж, таких як RNN, CNN та BERT. Сформульовано функціональні та нефункціональні вимоги до системи.

Розроблено діаграми використання, розгортання, послідовності та компонентів, які описують архітектуру системи. Спроектовано архітектуру нейронної мережі для класифікації тональності коментарів. Детально описано методи, які використовувалися для обробки тексту та класифікації тональності. Проведено тестування системи для оцінки її ефективності та точності. В результаті виконаної роботи було створено функціональну систему для аналізу тональності коментарів, яка використовує методи обробки природної мови та машинного навчання.

Система успішно класифікує коментарі на позитивні, нейтральні та негативні з високим рівнем точності, що підтверджується результатами тестування та оцінювання. Отримані результати можуть бути корисними для автоматизації аналізу настроїв користувачів у різних прикладних сферах, таких як соціальні мережі, маркетинг та обслуговування клієнтів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Feldman, R. (2013). Techniques and applications for sentiment analysis. Communications of the ACM, 56(4), 82-89. https://doi.org/10.1145/2436256.2436274
2. Liu, B. (2012). Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies, 5(1), 1-167. https://doi.org/10.2200/S00416ED1V01Y201204HLT016
3. Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2(1-2), 1-135. https://doi.org/10.1561/1500000011
4. Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4), e1253. https://doi.org/10.1002/widm.1253
5. Yadav, A., & Vishwakarma, D. K. (2020). Sentiment analysis using deep learning architectures: A review. Artificial Intelligence Review, 53(6), 4335-4385. https://doi.org/10.1007/s10462-019-09794-5
6. Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. IEEE Computational Intelligence Magazine, 13(3), 55-75. https://doi.org/10.1109/MCI.2018.2840738
7. SentimentAnalysis:ADefinitiveGuide [Електронний ресурс] - Режим доступу до ресурсу: https://monkeylearn.com/sentiment-analysis/ 4. SentimentAnalysis:Types,Tools,andUseCases. URL: https:// www.altexsoft.com/blog/business/sentiment-analysis-types-tools-a nd use-cases/
8. SentimentAnalysisinBanking–4CurrentUse-Cases/ URL: https://emerj.com/ai-sector overviews/sentiment-analysis-banking/
9. Аналіз тональності тексту [Електронний ресурс]- Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Аналіз_тональності_тексту
10. Вступ до машинного навчання. URL: http://specials.kunsht.com.ua/machinelearning2

11. BERT (модель мови). URL: https://uk.wikipedia.org/wiki/BERT_(модель_мови)
12. BERT Explained: State of the art language model for NLP. URL: https://towardsdatascience.com/ bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270
13. Word2vec. URL: https:// ru.wikipedia.org/wiki/Word2vec
14. Рекурентна нейронна мережа. URL: https://uk.wikipedia.org/wiki/ Рекурентна_нейронна_мережа
15. Text Classification with RNN. URL: https://towardsai.net/p/deep-learning/text-classification- with-rnn
16. Al Awar, S. (2023). Sentiment Analysis on News Headlines: Classic Supervised Learning vs Deep Learning Approach. Towards Data Science. URL: https://towardsdatascience.com/sentiment-analysis-on-news-headlines-classic-supervised-learning-vs-deep-learning-approach-5c4a59d6b3ce
17. Donmez, P., Lebanon, G., & Balasubramanian, K. (2010). Unsupervised supervised learning I: Estimating classification and regression errors without labels. Journal of Machine Learning Research, 11, 1323-1351.
18. Liu, B. (2020). Semi-supervised learning for sentiment classification using small number of labeled data. Journal of Data Science, 18(4), 673-691. https://doi.org/10.6339/20-JDS960
19. Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction (2nd ed.). MIT Press.
20. Wilbur, W. J., & Sirotkin, K. (1992). The automatic identification of stop words. Journal of Information Science, 18(1), 45-55. https://doi.org/10.1177/016555159201800106
21. Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. O'Reilly Media.
22. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781.

23. Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. IEEE Computational Intelligence Magazine, 13(3), 55-75. https://doi.org/10.1109/MCI.2018.2840738
24. Zhang, L., Wang, S., & Liu, B. (2018). Deep learning for sentiment analysis: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4), e1253. https://doi.org/10.1002/widm.1253
25. Ghosh, A., & Bakshi, S. (2014). Rule-based sentiment analysis for text data. In Proceedings of the 2014 international conference on advances in computing, communications and informatics (ICACCI) (pp. 1650-1654). https://doi.org/10.1109/ICACCI.2014.6968338
26. Baeldung. (2021). Requirements: Functional vs. Non-functional. Baeldung on Computer Science. https://www.baeldung.com/cs/functional-vs-non-functional-requirements
27. Introducing Types of UML Diagrams. URL: https://www.lucidchart.com/blog/types-of-UML-diagrams

ДОДАТОК А

```
,
import json

# Дані посту і коментарів
posts = [
    {
        "post_id": "1",
        "post_text": "Kevin De Bruyne is one of the greatest playmakers in PL history □□\n\n▪ 112 assists (2nd all-time)\n▪ 10+ assists in six different seasons\n▪ 10 assists in just 17 games this year",
        "comments": [
            "greatest player to ever grace the premier league.",
            "I don’t know why Raheem is included in this picture, Kevin could have been 60+ assist up of Raheem converted all those chances. The guy missed like he was competing with Nunez",
            "KVD > > > Magician of football world□□",
            "It should be Greatest not one of the Greatest.",
            "Top 3 greatest midfielders of all time",
            "He is one of the greatest ever in footballs history □",
            "But people will argue with me that Ozil is clear smh",
            "You will always stand as a legend to me",
            "HONESTLY SHE IS SO GOOD □□□□□□□□□□",
            "GOAT premier league midfielder and it's not even close",
            "Without him halaand is nothing"
        ]
    }
]

# Збереження даних у файл JSON
with open('posts_and_comments.json', 'w') as f:
    json.dump(posts, f, indent=4)

print("comments.json")
```

ДОДАТОК Б

```
import json
import numpy as np
import matplotlib.pyplot as plt

# Завантаження класифікованих коментарів
with open('classified_comments.json', 'r') as f:
    classified_comments = json.load(f)

# Ініціалізація змінних для підрахунку сум ймовірностей
total_probabilities = np.zeros(3)
count = 0

# Підрахунок сум ймовірностей для кожного класу
for comment in classified_comments:
    probabilities = np.array(comment['probabilities'][0])
    total_probabilities += probabilities
    count += 1

# Обчислення середніх ймовірностей для кожного класу
average_probabilities = total_probabilities / count

# Визначення загальної емоційної забарвленості поста
sentiment_mapping = {0: 'negative', 1: 'neutral', 2: 'positive'}
overall_sentiment                                                =
sentiment_mapping[np.argmax(average_probabilities)]

print(f"Average Probabilities: {average_probabilities}")
print(f"Overall Sentiment: {overall_sentiment}")

# Виведення кругової діаграми співвідношення настроїв
labels = ['Negative', 'Neutral', 'Positive']
colors = ['red', 'grey', 'green']
plt.figure(figsize=(8, 6))
```

```
plt.pie(average_probabilities, labels=labels, colors=colors,
autopct='%1.1f%%', startangle=140)
plt.title('Average Sentiment Probabilities')
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as
a circle.
plt.show()
```

ДОДАТОК В

```
import json
from transformers import AutoTokenizer,
AutoModelForSequenceClassification
import torch

# Завантаження моделі та токенізатора з локального сховища
def load_local_model():
    tokenizer =
AutoTokenizer.from_pretrained('./local_twitter_roberta')
    model =
AutoModelForSequenceClassification.from_pretrained('./local_twitter_
roberta')
    return tokenizer, model

# Функція для передбачення тональності
def predict_sentiment(tokenizer, model, text):
    inputs = tokenizer(text, return_tensors='pt', truncation=True,
padding=True)
    outputs = model(**inputs)
    probabilities = torch.nn.functional.softmax(outputs.logits, dim=-
1)
    sentiment = torch.argmax(probabilities, dim=-1).item()
    return sentiment, probabilities



# Завантаження коментарів з JSON
with open('posts_and_comments.json', 'r') as f:
    posts = json.load(f)

# Завантаження моделі та токенізатора
tokenizer, model = load_local_model()

# Класифікація коментарів з аналізом ймовірностей
results = []
sentiment_mapping = {0: 'negative', 1: 'neutral', 2: 'positive'}

for post in posts:
    for comment in post['comments']:
        sentiment, probabilities = predict_sentiment(tokenizer,
model, comment)
        sentiment_label = sentiment_mapping[sentiment]
        results.append({
            'post_id': post['post_id'],
```

```
            'comment': comment,
            'sentiment': sentiment_label,
            'probabilities': probabilities.detach().numpy().tolist()
        })

# Збереження результатів у JSON форматі
with open('classified_comments.json', 'w') as f:
    json.dump(results, f, indent=4)

print("Classified comments saved to classified_comments.json")
```

ДОДАТОК Г

Текст 1

1. "This is such a beautiful picture! Love the colors! □"
2. "Not really my taste, but I appreciate the effort."
3. "Wow, this is absolutely stunning! Well done! □"
4. "I've seen better from you, but still nice."

Текст 2

1. "Amazing post! You always inspire me! □"
2. "Hmm, this one didn't really resonate with me."
3. "Keep up the great work, love following your journey!"
4. "This is okay, but I think you can do better."

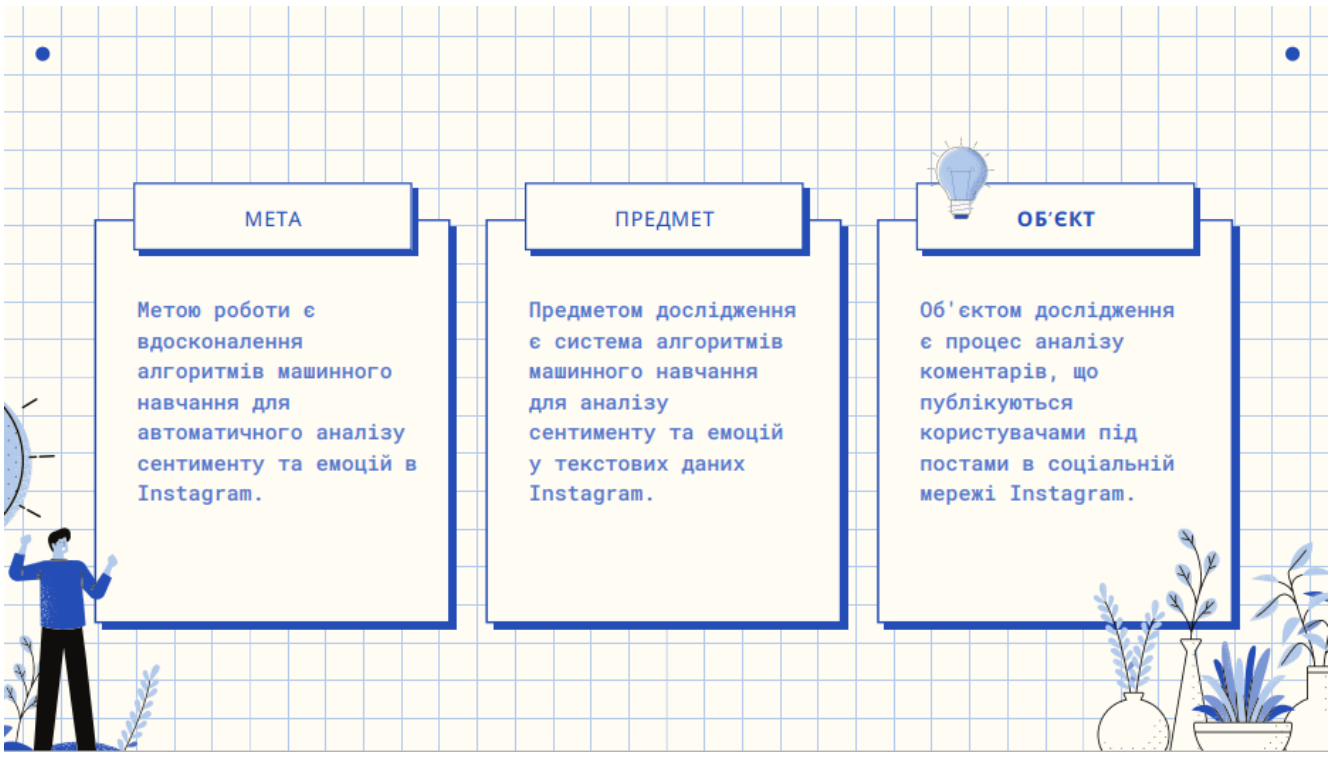
Текст 3

1. "Your content is always so refreshing! Thank you!"
2. "I don't get the hype around this, honestly."
3. "Such a creative idea! Love how you executed it!"
4. "This looks a bit off, maybe try a different angle next time?"

Текст 4

1. "You always bring a smile to my face with your posts! □"
2. "Sorry, but this isn't your best work."
3. "Fantastic shot! The lighting is perfect!"
4. "Not bad, but I feel like something is missing."

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ЗАВДАННЯ

ЗАВДАННЯМИ РОБОТИ Є:

1. АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ І МОДЕЛЕЙ ДЛЯ АНАЛІЗУ СЕНТИМЕНТУ ТА ЕМОЦІЙ.
2. РОЗРОБКА АЛГОРИТМУ АНАЛІЗУ ТОНАЛЬНОСТІ КОМЕНТАРІВ В INSTAGRAM.
3. ЗАСТОСУВАННЯ ПЕРЕДНАВЧЕНОЇ МОДЕЛІ ДЛЯ АНАЛІЗУ КОМЕНТАРІВ.
4. ОЦІНКА ЕФЕКТИВНОСТІ ЗАПРОПОНОВАНИХ МЕТОДІВ.

ТЕХНОЛОГІЇ

PyTorch

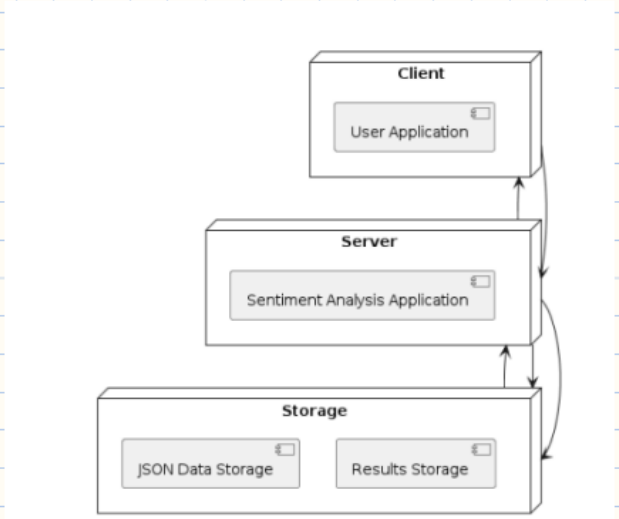
Hugging Face

JSON

NLP

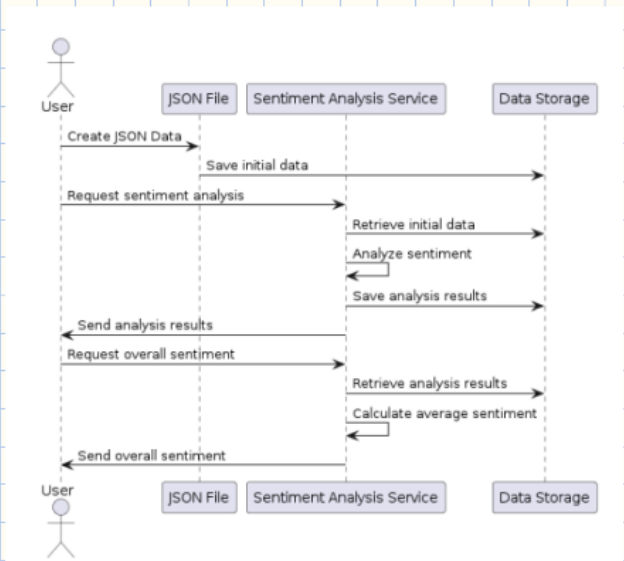
ДІАГРАМА ВИКОРИСТАННЯ

Діаграма використання (Use Case Diagram) є одним з типів діаграм UML (Unified Modeling Language), що використовуються для моделювання функціональних вимог системи.
Вона показує взаємодію між користувачами (акторами) та системою через набір випадків використання (use cases).
Основна мета такої діаграми – візуалізація того, як різні користувачі взаємодіють із системою та які функції вони використовують.



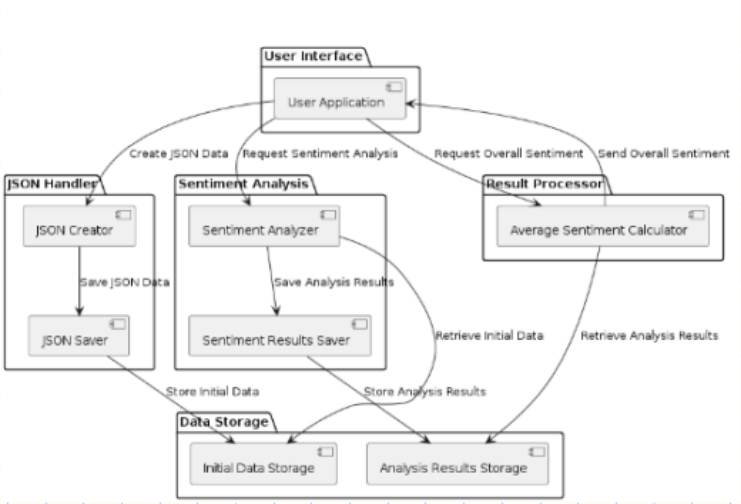
ДІАГРАМА РОЗГОРТАННЯ

Client - вузол, де користувач запускає програму для створення JSON даних та відправки запитів на сервер для аналізу.
Server - вузол, де виконується програма для аналізу тональності коментарів. Сервер отримує дані від клієнта, обробляє їх і зберігає результати.
Storage - вузол для зберігання даних, включаючи початкові дані JSON (пости та коментарі) та результати аналізу.



ДІАГРАМА ПОСЛІДОВНОСТІ

Діаграма послідовності (Sequence Diagram) є типом діаграми UML, яка показує взаємодію між об'єктами в системі у хронологічному порядку. Вона відображає послідовність повідомлень, що передаються між об'єктами для виконання певної функції або процесу.



ДІАГРАМА КОМПОНЕНТІВ

Діаграма компонентів (Component Diagram) є типом діаграми UML, яка показує організацію та залежності між програмними компонентами в системі.

ТЕСТУВАННЯ

СТВОРЕННЯ JSON ФАЙЛУ

Перейдемо до тестування системи. Для початку створимо файл JSON з коментарями до посту. Для цього запустимо файл comm.py.


```
"post_id": "1",
"post_text": "Kevin De Bruyne is one of the greatest playmake
"comments": [
    "greatest player to ever grace the premier league.",
    "I don\u2019t know why Raheem is included in this picture
    "KVD > > > Magician of football world",
    "It should be Greatest not one of the Greatest.",
    "Top 3 greatest midfielders of all time",
    "He is one of the greatest ever in footballs history ",
    "But people will argue with me that Ozil is clear smh",
    "You will always stand as a legend to me",
    "GOAT premier league midfielder and it's not even close",
    "HONESTLY SHE IS SO GOOD ",
    "Without him halaand is nothing"
]
```


ТЕСТУВАННЯ

АНАЛІЗ ТОНАЛЬНОСТІ

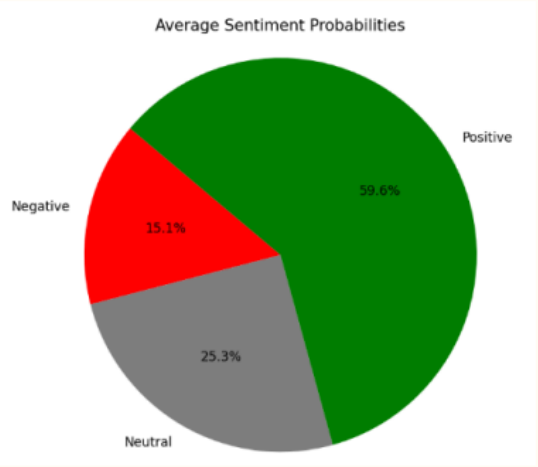
Наступним кроком є аналіз тональності всіх коментарів, для цього запустимо файл analize.py. Дані зберігаються у json файлі.


```
{
    "post_id": "1",
    "comment": "It should be Greatest not one of the Greatest.",
    "sentiment": "neutral",
    "probabilities": [
        [
            0.25092369318008423,
            0.6003116369247437,
            0.1487647145986557
        ]
    ]
},
```

ТЕСТУВАННЯ

СЕРЕДНЯ ТОНАЛЬНІСТЬ

Наступним кроком є визначення загальної тональності коментарів під постом, для цього запустимо файл avg.py. Отримуємо діаграму з розподіленням тональності та загальний результат.



ТЕСТУВАННЯ

СЕРЕДНЯ ТОНАЛЬНІСТЬ

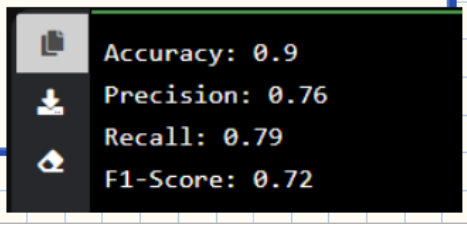
Це середні ймовірності для кожного з класів (негативний, нейтральний, позитивний) на основі аналізу тональності коментарів.
Overall Sentiment: positive - це загальна емоційна забарвленість поста на основі аналізу всіх коментарів. У цьому випадку загальна тональність визначена як позитивна, оскільки середня ймовірність для позитивного класу є найвищою серед усіх класів (59.60%).

```
Average Probabilities: [0.15148008 0.25256835 0.59595158]
Overall Sentiment: positive
```

ТЕСТУВАННЯ

МЕТРИКИ

- Accuracy (Точність): 0.90 - це означає, що 90% усіх передбачень моделі були правильними. Це осить високий показник, що вказує на загальну ефективність моделі.
- Precision (Точність): 0.76 - це означає, що 76% передбачених позитивних результатів були правильними. Тобто, з усіх випадків, коли модель передбачила позитивний результат, 76% дійсно були позитивними. Це показує, що модель має певну кількість хибнопозитивних результатів.
- Recall (Повнота): 0.79 - це означає, що модель правильно ідентифікувала 79% всіх фактичних позитивних випадків. Іншими словами, з усіх фактичних позитивних випадків модель правильно ідентифікувала 79%. Результат показує, що модель має певну кількість хибнонегативних результатів.
- F1-Score є гармонійним середнім між Precision і Recall. Значення 0.72 вказує на те, що модель має збалансовану ефективність між точністю і повнотою, хоча і не ідеальну. Це важливо в ситуаціях, коли потрібно балансувати між двома метриками, і коли високий Precision або Recall поодинці не є достатнім.



ВИСНОВКИ

Розроблена система успішно класифікує коментарі на позитивні, нейтральні та негативні з високим рівнем точності, що підтверджується результатами тестування та оцінювання. Отримані результати можуть бути корисними для автоматизації аналізу настроїв користувачів у різних прикладних сферах, таких як соціальні мережі, маркетинг та обслуговування клієнтів.