Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук
(повна назва)

Кафедра програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів аналізу сентименту на основі неструктурованих текстових відгуків
(тема)

Виконав:
студент (ка) 2 курсу, групи ІПЗм-22-6

Васильєв А. Р.
(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва спеціальності)
Тип програми освітньо-наукова

Керівник доц. Турута О. П.
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри _____ З.В.Дудар
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерних наук
Кафедра програмної інженерії
Рівень вищої освіти другий (магістерський)
Спеціальність 121 – Інженерія програмного забезпечення
Тип програми освітньо-наукова програма
Освітня програма Інженерія програмного забезпечення
(шифр і назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«___» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Васильєву Артему Ростиславовичу
(прізвище, ім’я, по батькові)

1. Тема роботи «Дослідження методів аналізу сентименту на основі неструктурованих текстових відгуків»

Затверджена наказом по університету від «29» березня 2024 р. №250 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 17.06.2024 р

3. Вихідні дані до роботи Дослідження методів аналізу сентименту на основі неструктурованих текстових відгуків

4. Перелік питань, що потрібно опрацювати в роботі

Метою роботи – аналіз сентименту в неструктурованих текстових відгуках, що становлять суттєву частину великої кількості інформації, що циркулює в мережі. Основний акцент роботи спрямований на розгляд методів, які дозволяють автоматично визначати та класифікувати емоційний тон виражених відгуків

КАЛЕНДАРНИЙ ПЛАН

№	Назви етапів курсової роботи	Термін виконання етапів роботи	Примітка
1	Видача теми, узгодження і затвердження	02.04.2024	виконано
2	Аналіз предметної галузі	02.04.2024	виконано
3	Огляд існуючих методів	09.04.2024	виконано
4	Оформлення пояснювальної записки	05.05.2024	виконано
5	Здача готового проекту	20.06.2024	виконано

Дата видачі завдання 20 січня 2024 р.

Студент  Васильєв А. Р.
(підпис) (прізвище, ініціали)

Керівник кваліфікаційної роботи _____ доц. Турута О. П.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка: 57 с., 12 рис., 5 додатків, 12 джерел.

МАШИННЕ НАВЧАННЯ, НЕЙРОННІ МЕРЕЖІ, СЕНТИМЕНТНИЙ АНАЛІЗ, СТАТИСТИЧНІ МЕТОДИ, ТЕКСТОВІ ВІДГУКИ.

Об'єктом дослідження є аналіз сентименту в неструктурованих текстових відгуках, що становлять суттєву частину великої кількості інформації, що циркулює в мережі. Основний акцент роботи спрямований на розгляд методів, які дозволяють автоматично визначати та класифікувати емоційний тон виражених відгуків.

Область дослідження включає в себе розгляд різних аспектів аналізу сентименту, таких як виявлення позитивних, негативних та нейтральних відгуків, визначення емоційних відтінків та інтенсивності виражених емоцій.

Дослідження також охоплює розгляд прикладних аспектів застосування методів аналізу сентименту у практичних областях, таких як маркетинг, соціальні дослідження, виробництво та інші, з метою визначення переваг та обмежень використання різних підходів у конкретних сценаріях.

Об'єкт дослідження, який базується на неструктурованих текстових відгуках, становить важливий внесок у розвиток області аналізу сентименту та відкриває нові можливості для вдосконалення технік обробки природної мови та інтелектуального аналізу текстів у сучасному інформаційному суспільстві.

TEXTUAL REVIEWS, SENTIMENT ANALYSIS, MACHINE LEARNING, NEURAL NETWORKS, STATISTICAL METHODS

The object of the study is sentiment analysis in unstructured textual reviews, which constitute a significant part of the vast amount of information circulating on the internet. The primary focus of the work is on examining methods that enable the automatic identification and classification of the emotional tone expressed in reviews.

The research area includes various aspects of sentiment analysis, such as detecting positive, negative, and neutral reviews, determining emotional shades, and the intensity of expressed emotions. Special attention is given to the development and comparison of machine learning algorithms, natural language processing, and deep learning to achieve the highest accuracy and efficiency in sentiment detection.

The study also covers the practical application aspects of sentiment analysis methods in areas such as marketing, social research, manufacturing, and others, aiming to determine the advantages and limitations of using different approaches in specific scenarios.

The research object, based on unstructured textual reviews, makes a significant contribution to the development of sentiment analysis and opens up new opportunities for improving natural language processing techniques and intelligent text analysis in the modern information society.

Я, Васильєв Артем Ростиславович, студент гр. ІПЗм-22-6, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя робота на тему «Дослідження методів аналізу сентименту на основі неструктурованих текстових відгуків», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу ElArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений(на) з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

Вступ 7
1 Дослідження методів аналізу сентименту на основі неструктурованих текстових відгуків 8
1.1 Виявлення ступінь вдоволення коментатора 8
1.2 Визначення емоційних відтінків та ступеню вираження емоцій 9
1.3 Порівняння методів обробки природньої мови для досягнення найвищої точності виявлення сентименту 11
2 Аналіз підходів до аналізу сентименту 15
2.1 Машинне навчання 15
2.1.1 Логістична регресія та SVM аналіз 18
2.2 Глибоке навчання 21
2.3 Лінгвістичний підхід 23
3 Проведення дослідження ефективності підходів 25
3.1 Порівняння ефективності машинного навчання та лінгвістичного методу. 26
3.2 Vader vs TextBlob 29
4 Розробка органічного поєднання двох підходів 35
4.1 Розробка стратегії поєднання 35
4.1.1 Ключова ідея 36
4.2 Розробка системи класів 38
Висновки 42
Перелік джерел посилання 43
Додаток А Перелік джерел посилання за науковими напрямками керівника та науковців кафедри програмної інженерії 45
Додаток Б Слайди презентації 46
Додаток В Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ 55
Додаток Г Апробація результатів роботи 56
Додаток Д Експертний висновок результатів перевірки кваліфікаційної роботи на відповідність оформлення вимогам ДСТУ 3008:2015 57

ВСТУП

В сучасному цифровому світі велика кількість інформації зберігається у вигляді неструктурованих текстових даних, зокрема відгуки користувачів у Інтернеті. Зростання популярності соціальних мереж, онлайн-магазинів та форумів призводить до необхідності розвинення та вдосконалення методів аналізу сентименту для отримання цінної інформації з цих неструктурованих джерел.

Об'єктом даного дослідження є методи аналізу сентименту, які базуються на неструктурованих текстових відгуках. Мета курсової роботи – дослідити та порівняти різноманітні підходи до аналізу сентименту, зокрема методи машинного навчання, обробки природної мови та глибокого навчання. Вивчення цих методів дозволить визначити їхні переваги та недоліки у контексті виявлення та класифікації емоційних відгуків.

Практичне значення дослідження полягає в можливості застосування ефективних методів аналізу сентименту для автоматичного оброблення великої кількості текстових даних у реальному часі. Такий аналіз може бути використаний у сферах маркетингу, бізнесу та соціальних досліджень для отримання важливої інформації про відгуки користувачів та сприйняття їхніх емоційних реакцій.

1 ДОСЛІДЖЕННЯ МЕТОДІВ АНАЛІЗУ СЕНТИМЕНТУ НА ОСНОВІ НЕСТРУКТУРОВАНИХ ТЕКСТОВИХ ВІДГУКІВ

1.1 Виявлення ступінь вдоволення коментатора

Виявлення ступеня вдоволення чи невдоволення коментатора на основі його текстового відгуку є актуальною та важливою темою у сучасному суспільстві. Особливо корисним така можливість є у таких сферах як електронна комерція, маркетинг і реклама, соціальні мережі, соціальні дослідження та політичний аналіз.

Врахування контексту є ключовим аспектом аналізу сентименту, оскільки емоційний тон часто залежить від сполучення слів та їх контекстуального вживання. Для досягнення ефективності виявлення сентименту необхідно враховувати семантичні та синтаксичні зв'язки між словами та фразами.

Семантичний аналіз включає в себе розгляд значень слів та їх взаємодії у конкретному контексті. Наприклад, певне слово може мати різне значення в залежності від того, як воно використовується в реченні. Визначення семантичних полів допомагає уточнити емоційний відтінок.

Синтаксичний аналіз включає розгляд структури речення та взаємозв'язків між його частинами. При цьому важливо враховувати, як зміна порядку слів чи структури речення може вплинути на загальний сентимент.

Контекстуальний аналіз передбачає врахування широкого контексту, включаючи тему обговорення та специфіку мовлення. Наприклад, термін "холодний" може мати різні емоційні конотації в залежності від теми – в погоді чи у відносинах.

Не менш важливим аспектом виявлення ступеня вдоволення коментатора є аналіз зміни тону у тексті. Цей аспект є важливим для виявлення динаміки емоційного стану коментатора впродовж усього тексту коментаря. Це включає розгляд того, як змінюється сентимент від речення до речення чи в межах усього відгуку.

Для аналізу зміни тону можна використовувати методи обробки часових рядів, які дозволяють виявляти плавні зміни настрою в тексті. Це корисно для

розуміння емоційного розвитку відгуку та виявлення конкретних моментів, які спричиняють зміни в сентименті. У процесі такого аналізу важливо враховувати контекстуальні фактори, такі як події, які можуть вплинути на емоційний стан коментатора. Наприклад, зміна настрою може бути пов'язана зі зміною теми обговорення чи виникненням нових обставин[1].

Для покращення ефективності аналізу можна використати нейронні мережі. Нейронні мережі, зокрема рекурентні та прямі нейронні мережі, здатні до глибокого аналізу текстової інформації. Вони автоматично вивчають складні зв'язки між словами та фразами, що дозволяє вдосконалити точність виявлення сентименту. Глибокі моделі, такі як рекурентні нейронні мережі та Long Short-Term Memory (LSTM) мережі, можуть враховувати довгострокові залежності в тексті, що особливо важливо для виявлення сентименту, що розвивається вздовж тексту коментаря користувача.

Також нейронні мережі можуть застосовуватися для контекстуального аналізу, адаптуючись до специфіки кожного конкретного відгуку та враховуючи індивідуальні властивості мовлення коментатора. Загальний аналіз врахування контексту, аналізу зміни тону та застосування нейронних мереж дозволяє створити комплексний підхід для виявлення ступеня вдоволення чи невдоволення коментатора на основі його текстового відгуку. Це інструменталізму надійний та точний аналіз емоційного відтінку у цифровому середовищі.

1.2 Визначення емоційних відтінків та ступеню вираження емоцій

Визначення емоційних відтінків та ступеню їх вираження у людей є предметом інтенсивних досліджень у галузі психології, когнітивної науки та обробки природної мови. Ця область вивчення зорієнтована на розуміння та аналіз людських емоцій з метою покращення комунікації, розробки технологій штучного інтелекту, а також розширення можливостей у багатьох інших сферах.

Емоції, як складова частина людського психічного життя, завжди привертала увагу вчених та дослідників. Існує велика різноманітність емоцій, від радості та задоволення до суму та розчарування, і кожна з них може мати різний

емоційний відтінок та ступінь вираження емоції. Розуміння цих відтінків та їх міри – ключовий аспект в розвитку нашого загального уявлення про людські емоції.

Емоційні відтінки визначаються як особливості вираження емоцій, які можуть бути більш чіткими та конкретизованими, ніж самі базові емоції. Основні емоційні відтінки можна поділити на дві категорії – позитивні відтінки та негативні відтінки, вони включають в себе наступні емоції:

Позитивні відтінки:

– радість – емоційний стан високого задоволення та позитивного настрою, часто викликаний приємними подіями чи ситуаціями;
– щастя – глибокий та тривалий стан позитивного емоційного комфорту, що супроводжується внутрішнім спокоєм та вдячністю;
– задоволення – емоційний стан комфорту та вдоволення, часто спровокований приємними подіями чи ситуаціями.

Негативні відтінки:

– сумнів – відчуття невпевненості чи непевності, що супроводжується роздумами та переживаннями;
– тривога – природне відчуття страху перед загрозою, яке дозволяє її помітити і вжити відповідні заходи[2];
– гнів – сильне не годування та невдоволення, часто викликане конфліктними ситуаціями чи неприємними подіями.

Аналіз природної мови є сучасним інструментом, який використовується для розуміння та вивчення емоційного відтінку в тексті. Застосовуючи алгоритми та методи Natural Language Processing, ми можемо визначити емоційне забарвлення конкретних слів, фраз і текстових відгуків. Техніки машинного навчання та обробки природної мови використовують синтаксичні та семантичні аналізи для виявлення емоційних нюансів, допомагаючи розкрити складність людського мовлення.

Текстовий аналіз спрямований на розгляд емоційного забарвлення у висловлюваннях людини. Дослідження використовує такі елементи, як

забарвлення тексту, контекст, емоційність, щоб визначити, які емоції виражені у коментарі. Експресивне письмо також є ефективним методом вивчення емоцій, оскільки воно дозволяє людям висловлювати свої внутрішні почуття та переживання. Аналізуючи виразність слів та вибір мовленнєвих засобів, дослідники можуть здобути інсайти в емоційний стан людини.

Ці лінгвістичні методи не лише виявляють факт емоційного відтінку, але і дозволяють аналізувати його виразність, інтенсивність та контекстуальні особливості. Такі підходи допомагають не лише кількісно вимірювати наявність емоцій, але і розуміти їхню глибоку природу та взаємозв'язок із змістом мовлення чи тексту.

1.3 Порівняння методів обробки природньої мови для досягнення найвищої точності виявлення сентименту

Обробка природньої мови (Natural Language Processing, NLP) стала ключовим напрямком в розробці систем для виявлення сентименту у текстових відгуках. Існує великий вибір методів та підходів для досягнення максимальної точності в аналізі емоційного відтінку тексту. У цьому тексті розглянемо та порівняємо різні методи NLP, звертаючи увагу на їхні переваги, недоліки та можливості в досягненні високої точності у виявленні сентименту.

Перший підхід це використання правилових методів. Правилові методи включають в себе використання певних правил та шаблонів для визначення сентименту у тексті. Цей підхід може бути ефективним для виявлення явних емоційних відгуків, але виявляється менш ефективним при аналізі більш складних конструкцій та випадків. Перевагами цього метода є простота реалізації та зрозумілість. Але недоліками є низька адаптивність до нових форм висловлення емоцій та обмеженість у роботі з неявними або амбігвітними (двозначними) виразами.

Другий підхід це використання нейронних мереж та машинного навчання. Машинне навчання в NLP використовує алгоритми для тренування моделей на основі прикладів. Цей підхід включає в себе використання класифікаційних

алгоритмів, таких як наївний Баєсівський класифікатор чи методи глибокого навчання, наприклад, рекурентні нейронні мережі та векторні представлення слів. Перевагами використання нейронних мереж є здатність виявлення складних та неявних відтінків сентименту та висока адаптивність до різноманітних типів тексту.

Недоліками у свою чергу є те що тренування мереж вимагає великого обсягу даних для ефективного тренування, окрім того саме тренування вимагає серйозних обчислювальних потужностей. Також не можна ігнорувати наявність можливості вивчення помилкових асоціацій при наявності неправильно позначених даних[3].

Третій підхід це глибоке навчання та рекурентні нейронні мережі. Глибоке навчання у комбінації з нейронними мережами, зокрема RNN, здатні враховувати контекстуальні залежності в тексті та отримувати більше інформації про відтінки сентименту, які можуть змінюватися вздовж тексту. Перевагами використання глибокого навчання є покращена здатність розпізнавання довгострокових залежностей у тексті, та висока точність при аналізі складних синтаксичних конструкцій.

Однак як і у разі використання машинного навчання цей підхід потребує значних обчислювальних ресурсів та тривалих часових витрат для тренування, а також великих даних для можливості проведення процесу тренування мережі. Також глибоке навчання хоча і усуває деякі недоліки з якими ми маємо справу у випадку роботи з машинним навчанням, але відкриває нову вразливість до проблеми зникаючого та вибухаючого градієнту.

В глибокому навчанні, особливо при використанні глибоких нейронних мереж, існує два поширені явища, відомі як проблема зникаючого та вибухаючого градієнту. Ці явища можуть виникати під час тренування моделей і впливати на їхню ефективність та здатність до навчання.

Проблема зникаючого градієнту виникає, коли градієнти, які передаються назад у процесі зворотного поширення помилки, стають дуже малими. Це

особливо стосується глибоких мереж, де градієнти можуть експоненційно зменшуватися при кожному проходженні вглиб.

Це погано тому що коли градієнти стають дуже малими, нейронна мережа може втратити здатність до ефективного навчання. Ваги нейронів не оновлюються належним чином, і, отже, модель не може адаптуватися до нових даних.

Проблема вибухаючого градієнту виникає, коли градієнти стають дуже великими, особливо при використанні глибоких архітектур. Це може впливати на стійкість навчання та призводити до нестабільності моделі.

Занадто великі градієнти, можуть призводити до того, що оновлення ваг стають надто великими, і модель може не зберігати стабільний стан. Це може призводити до неадекватного навчання, втрати здатності до узагальнення та перенавчання.

Однак існують засоби мінімізації цих проблем. Декілька стратегій використовуються для мінімізації проблеми зникаючого та вибухаючого градієнту. Наведемо мінімізацію:

– нормалізація градієнту – використання методів нормалізації градієнту, таких як batch normalization, допомагає утримувати градієнти на прийнятному рівні, запобігаючи їхньому вибуханню чи зникненню;
– використання альтернативних функцій активації – вибір альтернативних функцій активації, може запобігти зниканню градієнту, дозволяючи непозитивні значення для нейронів;
– калібрація навчання параметрів мережі – грамотний вибір параметрів мережі, таких як швидкість навчання та ініціалізація ваг, може покращити стійкість моделі до проблем зникаючого та вибухаючого градієнту.

Проблеми зникаючого та вибухаючого градієнту є важливими аспектами глибокого навчання, які можуть впливати на ефективність тренування моделей. Розуміння цих явищ та використання відповідних стратегій може допомогти у покращенні стійкості та ефективності глибоких нейронних мереж.

Четвертим методом порівняння методів обробки природньої мови є використання векторних представлень слів. Word Embeddings, або векторне представлення слів, використовується для представлення слів у вигляді векторів у N-вимірному просторі, де подібні слова розташовані поруч. Це полегшує аналіз семантичних відносин та сприяє точнішому виявленню сентименту.

Перевагами цього методу є здатність враховувати семантичні зв'язки між словами. При цьому навідміну від попередніх методів які включали в себе використання нейронних мереж і необхідності їх навчання, цей метод є ефективним при роботі з обмеженими ресурсами. Основним недоліками методу є важкість у виявленні слів або виразів, які не входять в область векторних представлень.

Моделі, які використовуються для створення Word Embeddings, навчаються на певному обсязі текстів. Якщо слово або вираз не зустрічається у цьому масиві, то модель не може надати йому відповідного векторного представлення. Це особливо актуально для нішових або специфічних галузей, де можуть існувати терміни або термінологія, які не включені у загальні мовленнєві моделі. У таких випадках, системі слід буде додатково навчати або адаптувати на нових даних для ефективного використання Word Embeddings.

2 АНАЛІЗ ПІДХОДІВ ДО АНАЛІЗУ СЕНТИМЕНТУ

2.1 Машине навчання

Машинне навчання та обробка природної мови є двома важливими галузями в інформатиці і цьому дослідженні. Вони поєднують у собі теоретичні концепції та практичні застосування для створення інтелектуальних систем.

Машинне навчання є галуззю штучного інтелекту, що дозволяє комп'ютерам навчатися на основі даних та досвіду. Обробка природної мови вивчає методи обробки та аналізу мовлення та текстів на природній мові.

Обидві галузі є активними об'єктами дослідження та розвитку, і їх взаємодія призводить до створення розумних систем, здатних розуміти та взаємодіяти з людьми. Розглянемо ключові алгоритми, техніки та підходи, які використовуються в машинному навчанні та обробці природної мови.

Навчання з учителем (Supervised Learning) є одним з ключових підходів у машинному навчанні, де модель навчається на основі набору прикладів, який містить вхідні дані та відповідні мітки чи вихідні значення. Основна ідея полягає в тому, щоб навчити модель відображати вхідні дані на відповідні мітки, тобто вчиться відношенням між вхідними та вихідними змінними. Для використання цього підходу потрібно виконати ряд умов, а саме:

- визначення задачі, яку задачу потрібно розв'язати, чи це класифікація, регресія чи інше;
- зібрання набору даних, який містить пари вхідних даних та відповідних міток;
- розбиття набору даних на тренувальний, валідаційний та тестовий набори: це дозволяє оцінити ефективність моделі на нових, раніше не бачених даних;
- нормалізація та стандартизація: приведення вхідних даних до одного масштабу для полегшення навчання моделі;
- обробка відсутніх значень: обробка відсутніх або невірних даних для покращення якості моделі;

– вибір моделі або алгоритму, який найкраще відповідає задачі;
– визначення структури та параметрів моделі, також відомий як вибір архітектури[4];
– передача тренувальних даних моделі для навчання;
– оптимізація параметрів моделі, щоб мінімізувати функцію втрати;
– використання валідаційного набору для оцінки точності та уникнення перенавчання;
– використання тестового набору для оцінки загальної ефективності моделі.

Для самого навчання використовуються наступні алгоритми:

– лінійна регресія - застосовується для задач регресії, де потрібно передбачити числовий вихід на основі вхідних даних;
– метод опорних векторів - використовується для класифікації та регресії, знаходячи оптимальний гіперплощину для розділення класів;
– рішучі дерева та випадковий ліс - ефективні для класифікації та регресії, здатні моделювати нелінійні залежності в даних;
– нейронні мережі - використовуються для різноманітних завдань, включаючи великі дані та завдання зображень, мовлення та тексту.

Навчання без учителя (Unsupervised Learning) – це підхід в машинному навчанні, при якому модель працює з набором даних, де відсутні мітки або вихідні значення. Основна мета - вивчення структури та патернів в даних без явного вказівника, як класифікувати чи передбачати вихідні значення. Навчання без учителя використовується для завдань, таких як кластеризація, розмірність, тематичне моделювання та асоціативний аналіз. Для використання цього підходу потрібно виконати ряд умов, ці умови у більшості такі самі як і у навчання з учителем, а саме: збір даних, їх обробка, вибір і навчання моделі і тд.

Основною різницею навчання без учителя від навчання з учителем є алгоритми що ми використовуємо. У навчанні без учителя використовуються ми використовуємо дані які не мають поміток, як у навчанні з учителем. Основні алгоритми для навчання це:

– кластеризація k-середніх - групує дані в k кластерів, де кожен кластер має свої унікальні характеристики;
– алгоритми головних компонентів - зменшує розмірність даних шляхом відображення їх у простір меншої розмірності, зберігаючи при цьому максимальну дисперсію;
– автокодування - використовується для зменшення розмірності та вилучення патернів у даних;
– методи асоціативного аналізу - знаходження асоціацій та взаємозв'язків між елементами в наборі даних.

Нейромережі виконують ряд завдань при роботі із текстами, та визначення емоціонального забарвлення тексту.

Автоматичне розпізнання частин мови, одне із важливих завдань в обробці природної мови є автоматичне розпізнавання частин мови. Це включає визначення граматичної категорії (іменник, прикметник, дієслово тощо) для кожного слова у тексті. Алгоритми, такі як Hidden Markov Models (HMM) чи Conditional Random Fields (CRF), використовуються для ефективного вирішення цього завдання. POS-Tagging важливий для багатьох прикладних областей, таких як машинний переклад та аналіз текстів.

Аналіз сентименту визначає та класифікує емоційний тон тексту, часто вказуючи на позитивний, негативний чи нейтральний настрій. Класичні методи, такі як машинне навчання на основі байєсівських класифікаторів або метод опорних векторів, можуть використовуватися для реалізації аналізу сентименту. Це важливий інструмент для розуміння відгуків, виявлення настрою в соціальних мережах чи вирішення проблем у сфері обслуговування клієнтів.

Генерація тексту - це завдання, в якому система створює новий текст, який має схожий стиль чи контекст з вхідними даними. Марковські моделі чи глибокі рекурентні мережі можуть використовуватися для генерації тексу. Це застосовується у сферах створення контенту, генерації автоматичних відповідей чи підтримки творчого написання[5].

Екстракція іменованих сутностей - визначає та класифікує іменовані об'єкти у тексті, такі як імена людей, місця, організації тощо. Техніки, такі як використання моделей на основі правил, статистичних методів (наприклад зазначений раніше, Conditional Random Fields) або глибокого навчання, допомагають розв'язати це завдання. Це також можна використовувати для створення баз даних, покращення пошукових систем та аналізу новин[6].

Автоматичне розпізнавання залежностей. Автоматичне розпізнавання залежностей моделює синтаксичні зв'язки між словами у реченні. Це важливо для розуміння структури мовлення. Алгоритми, такі як алгоритми динамічного програмування або методи на основі глибокого навчання, можуть використовуватися для ефективного вирішення цього завдання.

Ці різноманітні методи обробки природної мови використовуються в різних сферах та допомагають розширювати можливості взаємодії комп'ютерів із людьми у величезному спектрі застосувань. Також варто зазначити що хоча ці завдання постійно виконуються при навчанні нейромереж, вони також можуть бути застосовані і без їх використання.

Наприклад їх можна комбінувати із іншими більщ традиційними методами машинного навчання. Особливо в задачах, де дані є не дуже обширними чи для завдань, де важлива ефективність та інтерпретованість. Для нашої задачі аналізу сентименту на не структурованих текстових відгуках, можна створити рішення використовуючи машинне навчання основане на байєсівських класифікаторах чи методі опорних векторів. При цьому традиційні алгоритми можуть використовувати рядок векторів та статистичні характеристики для класифікації тексту на позитивний, негативний чи нейтральний.

2.1.1 Логістична регресія та SVM аналіз

Логістична регресія та машини опорних векторів (SVM) є популярними методами для аналізу сентименту у машинному навчанні, оскільки вони ефективно справляються з класифікаційними задачами і мають декілька переваг:

Логістична регресія:

– простота розуміння та імплементації: логістична регресія – це порівняно простий алгоритм, який моделює ймовірність належності до класу за допомогою сигмоїдної функції. це дозволяє легко інтерпретувати результати, що є корисним у застосуваннях, де необхідне зрозуміле рішення;
– ефективність при великій кількості ознак: логістична регресія ефективно справляється з високо вимірними даними, що є типовим для задач nlp, де кожне слово може розглядатися як окрема ознака;
– хороша прогностична здатність: навіть за наявності порівняно простої лінійної моделі, логістична регресія може давати високу точність у задачах бінарної класифікації.

Машини опорних векторів (SVM):

– гнучкість: svm може моделювати як лінійні, так і нелінійні межі рішень, використовуючи різні ядра (наприклад, лінійне, поліноміальне, радіально-базисне та інші);
– мінімізація помилок: svm спрямовані на максимізацію маржі між класами, що може сприяти кращій загальній продуктивності та стійкості моделі до перенавчання, особливо в умовах обмежених даних;
– ефективність у складних просторах: svm часто використовуються у складних задачах класифікації, де звичайні лінійні моделі недостатньо ефективні.

Обидва методи мають сильні сторони в контексті обробки природної мови і зокрема аналізу сентименту, завдяки їхній здатності адекватно обробляти і класифікувати текстові дані. Тому вони широко застосовуються для розробки систем, які потребують здатності розуміти емоційний контекст тексту, таких як ваш дипломний проект, зосереджений на аналізі сентименту невпорядкованих текстових відгуків. На рисунку 2.1 наведено сценарій використання SVM у системі.

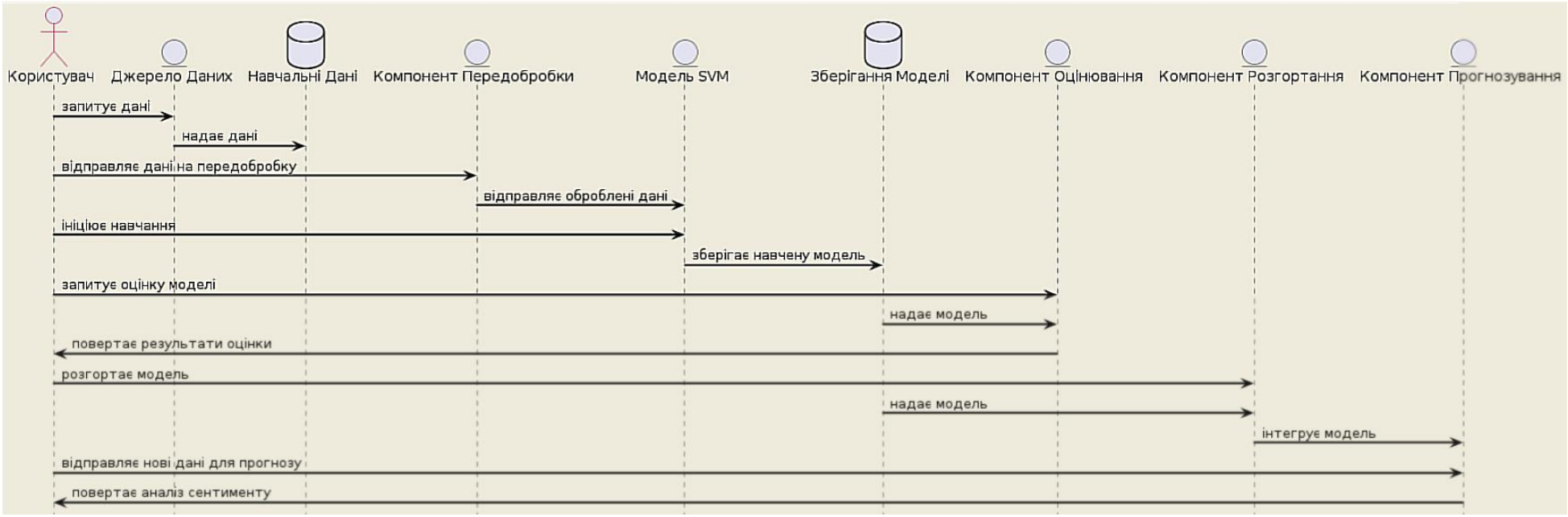

Рисунок 2.1.1 – сценарій використання SVM

2.2 Глибоке навчання

Глибоке навчання в аналізі сентименту дозволяє здійснити складний процес розуміння людської мови завдяки своїй здатності виділяти і обробляти суттєві особливості тексту. Зокрема, мережі згорткові (CNN) ефективно визначають важливі слова або фрази та аналізують локальні залежності у тексті, що впливають на сентимент[7]. Це стає можливим завдяки фокусуванню на специфічних словосполученнях через згорткові шари.

Рекурентні нейронні мережі (RNN) та їх удосконалення у вигляді LSTM (Long Short-Term Memory) мережі розширюють можливості аналізу, забезпечуючи збереження інформації про попередні слова і дозволяючи моделі краще розуміти контекст і послідовність елементів у тексті. Ще одним проривом є трансформери, зокрема модель BERT, яка використовує механізми уваги для зосередження на важливих словах, дозволяючи аналізувати текст у двох напрямках і тим самим глибше розуміти контекст та нюанси сентименту.

Весь процес аналізу сентименту розпочинається з попередньої обробки, де текст очищується від шуму і токенізується. Наступним кроком є векторизація, де текст перетворюється в числові вектори за допомогою таких методів як Word2Vec або GloVe, або через вбудовування токенів з предтренованих моделей.

Після векторизації відбувається навчання моделі на навчальних даних за допомогою обраної архітектури нейронної мережі, що пізніше оцінюється та тонко налаштовується для досягнення оптимальних результатів. Завдяки цим технологіям, глибоке навчання забезпечує високу точність і детальне розуміння тексту, відкриваючи нові можливості для аналізу сентименту.

Рекурентні нейронні мережі (RNN), LSTM (Long Short-Term Memory) і GRU (Gated Recurrent Unit) – це три важливих типи архітектур глибокого навчання, що широко використовуються для аналізу послідовних даних, таких як текст, часові ряди та інше. Кожна з цих архітектур має свої унікальні переваги:

Переваги RNN (Рекурентні нейронні мережі):

– обробка послідовностей даних: rnn ефективно обробляють дані, де порядок важливий. вони здатні "пам'ятати" інформацію про попередні елементи послідовності, що дозволяє їм враховувати контекст при обробці наступних елементів. це робить їх дуже корисними для задач, де контекст має значення, як, наприклад, при генерації тексту або розпізнаванні мови;
– простота моделі: rnn мають порівняно просту структуру, що складається з одного рекурентного шару, який може бути легко розширений або інтегрований з іншими типами мереж. це робить їх відносно легкими для розуміння та імплементації у порівнянні з більш складними архітектурами.

Переваги LSTM (Long Short-Term Memory):

– подолання проблеми зникнення градієнту: однією з ключових переваг lstm є їх здатність запобігати зникненню або вибуху градієнтів, проблемі, яка часто спостерігається в rnn. lstm використовують спеціальні шлюзові механізми, що дозволяють їм зберігати інформацію на тривалий час, не втрачаючи її важливість з плином часу;
– краща здатність до запам'ятовування: завдяки шлюзам забування та вводу, lstm можуть ефективно регулювати потік інформації, зберігаючи лише ту, що необхідна для виконання завдання. це дозволяє їм краще впоратися з дуже довгими послідовностями і складними задачами, де потрібне детальне розуміння контексту.

Переваги GRU (Gated Recurrent Unit):

– спрощена версія lstm: gru є спрощеною версією lstm з меншою кількістю параметрів, оскільки вона використовує два шлюзи (оновлення та скидання) замість трьох. це робить їх легшими для тренування табільш ефективними, коли потрібно швидше досягти результатів;

Ефективність у моделюванні залежностей: GRU відмінно справляються з моделюванням часових залежностей в даних, що робить їх дуже ефективними для задач, як-от прогнозування часових рядів і обробка мови. Шлюзи оновлення та скидання допомагають моделі вирішити, яку інформацію слід зберегти або

відкинути, що дозволяє GRU краще адаптуватися до змін у вхідних даних без значної втрати важливої інформації.

Обидва LSTM і GRU були розроблені для вирішення обмежень звичайних RNN, зокрема проблем зникнення градієнту, що робить їх важливими інструментами в наборі інструментів для глибокого навчання. Ці моделі виявляються особливо корисними у складних завданнях, де потрібно аналізувати великі масиви даних із сильними часовими або послідовними залежностями.

2.3 Лінгвістичний підхід

Лінгвістичний метод в аналізі тексту, зокрема в обробці природної мови, базується на використанні знань про мову для аналізу та розуміння тексту. Цей підхід залучає різні аспекти лінгвістики, включаючи синтаксис, семантику, морфологію та прагматику, щоб інтерпретувати та обробляти мовні дані. Ось декілька ключових аспектів лінгвістичного методу:

Синтаксичний аналіз: Цей процес включає аналіз структури речень, визначення граматичних ролей слова, і відносин між словами у реченні. Це допомагає системам зрозуміти, як слова поєднуються, щоб формувати значення.

Семантичний аналіз: Зосереджується на визначенні значення слів у контексті і як це значення взаємодіє з іншими словами для формування значень фраз або речень. Семантичний аналіз може включати розробку семантичних мереж, які показують взаємозв’язки та атрибути понять у тексті.

Морфологічний аналіз: Вивчення структури слів і використання морфем – найменших значущих одиниць мови. Це допомагає системам розпізнавати різні форми слова і правильно їх обробляти.

Програмні засоби: Лінгвістичні методи часто використовують спеціалізоване програмне забезпечення для аналізу мовних даних[8]. Це можуть бути інструменти для глибокого синтаксичного аналізу, семантичного моделювання, а також бібліотеки для обробки тексту, такі як NLTK у Python.

Примінення в різних областях: Лінгвістичний аналіз може застосовуватись в широкому спектрі завдань, включаючи машинний переклад, автоматичне резюмування, витягування інформації, аналіз сентименту та багато інших.

Лінгвістичний метод має свої переваги у точності та глибині обробки тексту, але також потребує значних ресурсів для розробки та підтримки складних лінгвістичних моделей. Цей підхід ідеально підходить для застосувань, де важливі глибоке розуміння тексту та висока точність інтерпретації.

3 ПРОВЕДЕННЯ ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ПІДХОДІВ

Ми вирішили не розглядати метод глибокого навчання при порівнянні методів аналізу сентименту з кількох причин. По-перше, методи глибокого навчання, хоч і ефективні, вимагають значних ресурсів для реалізації та обслуговування. Це включає потребу у великих обсягах даних для навчання моделей і високу обчислювальну потужність, яка часто реалізується через спеціалізоване обладнання, таке як GPU.

По-друге, глибоке навчання вимагає значних зусиль на етапі попередньої обробки даних та тонкої настройки параметрів моделі, що може бути складним і часозатратним. Необхідність у глибоких технічних знаннях для ефективного використання цих моделей може також обмежувати їхнє застосування в середовищах, де відсутні спеціалісти у галузі машинного навчання.

Крім того, складність моделей глибокого навчання ускладнює їх інтерпретацію та зрозуміння механізмів прийняття рішень, що може бути критично важливим у додатках, де необхідна прозорість та можливість пояснення результатів.

З огляду на ці аспекти, ми вирішили зосередитись на більш традиційних і менш ресурсо-містких методах, які дозволяють досягти гарних результатів при менших затратах та спрощують процес інтеграції та використання в різних проектах.

Отже надалі нами буде порівняно підходи більш прості в реалізації, а саме підхід машинного навчання та лінгвістичний метод.

Моделі глибокого навчання потребують більших обсягів даних; вони працюють краще, маючи доступ до звичайних даних. Навпаки, багато алгоритмів машинного навчання можуть давати задовільні результати навіть із меншими наборами даних. Це дуже важливо враховувати початківцям за даними при виборі між методологіями, особливо коли йдеться про обмеження доступності даних(див. рис. 3).

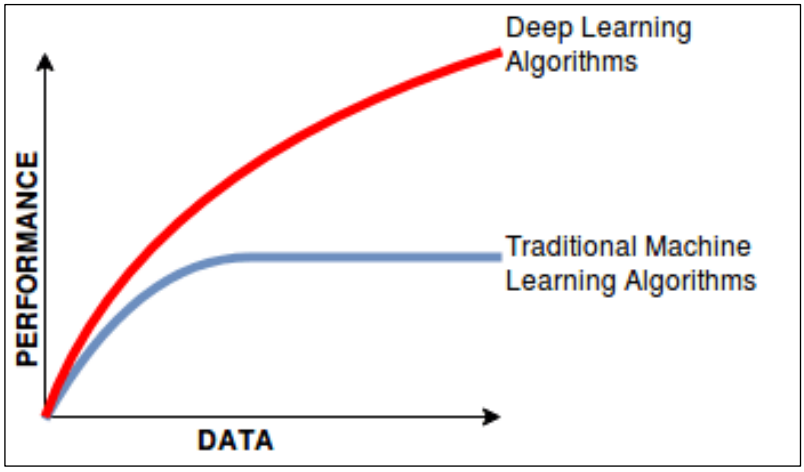

Рисунок 3 – Графік залежності обсягу даних (вісь X) від продуктивності (вісь Y)

На наведеному вище малюнку видно, що з збільшенням даних продуктивність алгоритмів глибокого навчання збільшується проти традиційним алгоритмом машинного навчання, у якому через деякий час продуктивність майже досягає насичення, навіть якщо дані збільшуються.

Відповідно для аналізу більшого обсягу даних – геометрично зростають вимоги до обчислювальних потужностей, отже приходиться більше ресурсів витрачати на GPU на відміну наприклад від машинного навчання для якого вистачає потужностей CPU[9].

3.1 Порівняння ефективності машинного навчання та лінгвістичного методу

Для початку слід розглянути детально плюси та мінуси обраних нами двох підходів, результати наведено у таблиці 3.1.

Таблиця 3.1 Переваги та недоліки методів

Метод	Переваги	Недоліки
Лінгвістичний аналіз	– точність у визначенні мовних особливостей: лінгвістичний метод дозволяє детально аналізувати синтаксис і семантику, використовуючи	– час та ресурси на розробку: розробка ефективних лінгвістичних систем може вимагати значного часу та експертних

Продовження таблиці 3.1

Метод	Переваги	Недоліки
	знання про мовні правила та структури. це може бути особливо корисним для застосувань, що вимагають глибокого розуміння мовних нюансів, наприклад, при перекладі або обробці складних інструкцій. – менша залежність від великих даних: на відміну від методів машинного навчання, багато лінгвістичних алгоритмів можуть ефективно працювати з меншими обсягами даних, оскільки вони базуються на чітко визначених мовних правилах.	знань у галузі лінгвістики та програмування. – обмежена гнучкість: лінгвістичні моделі часто вимагають ручного оновлення для адаптації до змін у використанні мови або для включення нових лінгвістичних феноменів, що може ускладнити їх використання в динамічних умовах.
Машинне навчання	– гнучкість та масштабованість: методи машинного навчання можуть автоматично адаптуватися до нових даних і масштабуватися для обробки великих обсягів інформації, що робить їх ідеальними для веб-середовища та додатків, що вимагають швидкої обробки великих обсягів даних.	– потреба в великих обсягах даних: більшість алгоритмашинне навчання може вимагати великих обсягів даних для ефективного тренування моделей, особливо для методів глибокого навчання. це може ускладнити використання у сферах, де доступ до великих даних обмежений або дорогий.

Кінець таблиці 3.1

Метод	Переваги	Недоліки
	– можливість виявлення складних закономірностей: сучасні алгоритми машинного навчання, зокрема глибоке навчання, можуть ідентифікувати складні шаблони і взаємозв'язки в даних, що може виявитися недосяжним для традиційних лінгвістичних підходів.	– висока обчислювальна потужність: методи глибокого навчання, такі як нейронні мережі, часто вимагають спеціалізованих обчислювальних ресурсів, наприклад gpu або tpu, що може зробити їх використання дорогим та технічно складним.

Отже загалом можна відмітити що машинне навчання та лінгвістичні методи мають різні сильні сторони у аналізі формальної та неформальної мови відповідно через особливості їх підходів до обробки тексту.

Машинне навчання віддає перевагу формальній мові, що наближена до стандартної, з кількох причин:

– чітка структура: формальна мова зазвичай дотримується граматичних правил і структур, які легше моделювати за допомогою алгоритмів машинного навчання. ці алгоритми можуть ефективно використовувати статистичні залежності та взаємодії між частинами мови, що є типовими для формальних текстів;
– залежність від великих даних: машинне навчання часто залежить від великих даних для тренування моделей. формальні тексти, такі як наукові статті, офіційні документи, часто мають великі обсяги та доступні у структурованому вигляді, що спрощує збір та обробку даних.

Лінгвістичні методи мають перевагу при аналізі неформальної мови через наступні аспекти:

- гнучкість правил: неформальна мова часто включає сленг, жаргон та ідіоми, які важко стандартизувати і моделювати через традиційні машинні методи. лінгвістичні підходи можуть включати більш гнучкі правила та адаптації, які краще впораються з мінливістю та особливостями неформальних текстів;
- контекстуальне розуміння: лінгвістичні методи можуть включати більш глибоке розуміння мовних нюансів, таких як іронія, сарказм або подвійне значення, що часто використовуються в неформальному спілкуванні. це дозволяє більш точно аналізувати сентимент або інтенції автора.

3.2 Vader vs TextBlob

Ми порівнюємо два підходи(машинне навчання та лінгвістичні моделі) на двох бібліотеках які їх реалізують vader vs textblop.Варто пояснити що ці бібліотеки є провідними бібліотеками з відкритим вихідним кодом для свого метода на даний час, тож ефективність методів в рамках дослідження можна прив’язати до ефективності бібліотек.

VADER і TextBlob є одними з найвідоміших та найбільш використовуваних бібліотек для аналізу сентименту з відкритим кодом. Вони представляють два різних підходи до розуміння тексту:

VADER (Valence Aware Dictionary and sEntiment Reasoner) є спеціалізованою бібліотекою, яка розроблена для роботи з соціальними медіа та іншими веб-текстами, де часто використовується неформальна мова, сленг та емодзі. Це лінгвістична модель, яка використовує набір заздалегідь визначених правил для аналізу сентименту[10].

TextBlob пропонує підхід, заснований на машинному навчанні, з використанням класифікаційних та регресійних алгоритмів. Ця бібліотека більш універсальна та придатна для широкого спектра застосувань від простого аналізу сентименту до складних задач обробки природної мови.

Для аналізу, представленого на графіку(див. рис. 3.3) у зображенні, було використано дата-сет відгуків про готелі, який містить думки клієнтів, які

зупинялися в готелі. Цей дата-сет включав текстові описи відгуків, які були аналізовані за допомогою бібліотек TextBlob та VADER для визначення сентименту цих текстів. Кожен відгук був оброблений для отримання оцінок полярності (від -1 до +1, де -1 означає негативний сентимент, а +1 – позитивний) та суб'єктивності (від 0 до 1, де 0 означає об'єктивний текст, а 1 – суб'єктивний).

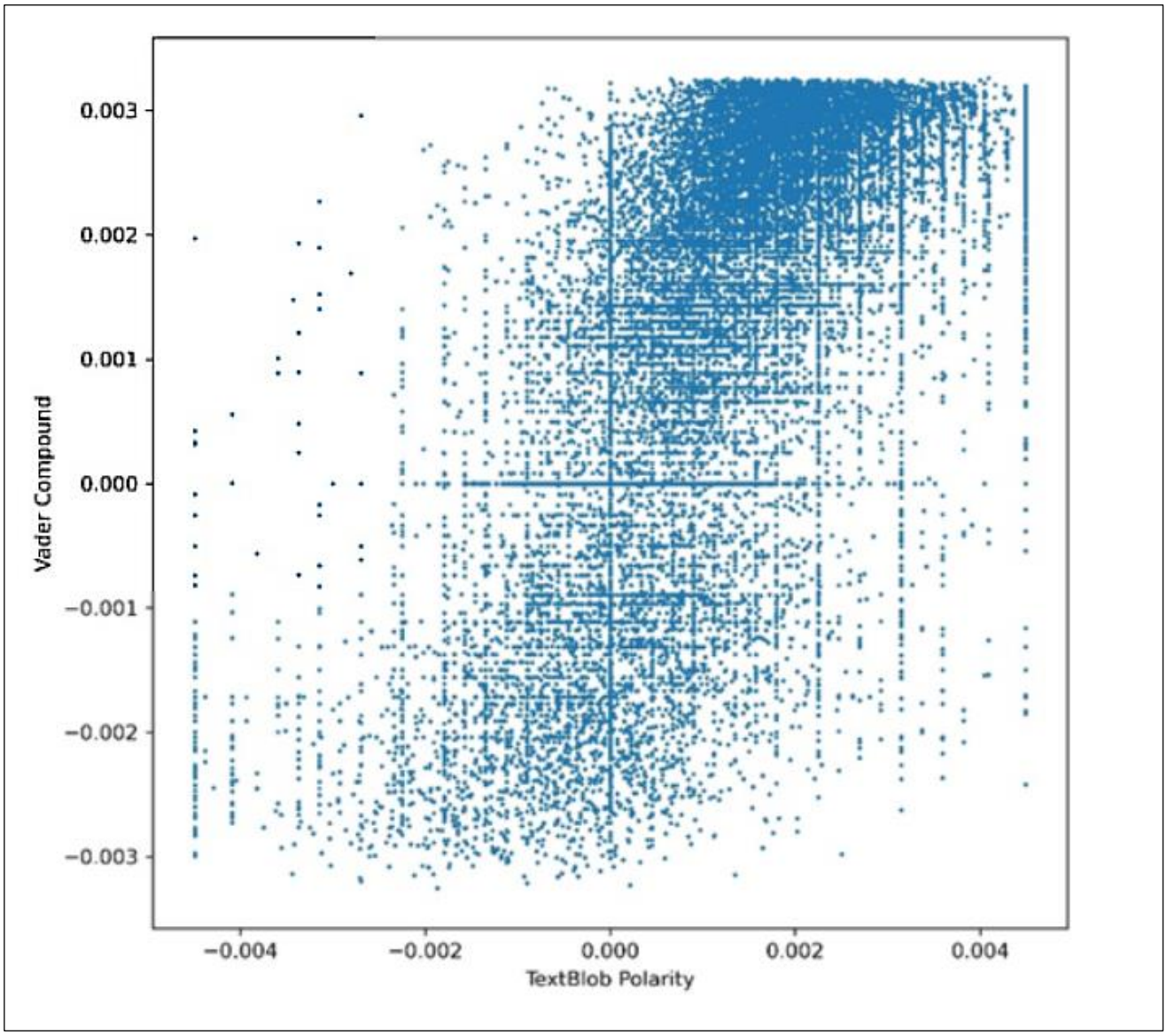

Рисунок 3.3 – Розподіл результатів аналізу[11]

Діаграма розсіювання вище ілюструє коефіцієнт кореляції Пірсона між VADER та TextBlob. Аналізуючи графік, бачимо, що хоча VADER класифікував певні пропозиції як негативні, TextBlob ідентифікував їх переважно як позитивні. У першому та третьому квадрантах обидва алгоритми показують згоду. Однак у другому та четвертому квадранті спостерігається помітна неузгодженість,

особливо у четвертому квадранті, де переважають суперечливі дані. TextBlob позначає пропозиції як позитивні, тоді як VADER класифікує їх як негативні.

Для аналізу неструктурованих текстових відгуків, особливо з соціальних медіа або інтернет-форумів, де часто зустрічаються сленг, емодзі та неформальна мова, VADER є однією з кращих бібліотек. VADER (Valence Aware Dictionary and sEntiment Reasoner) спеціально розроблена для роботи з такими типами текстів.

VADER є вибором для неструктурованих відгуків з наступних причин:

- спеціалізований на соціальних медіа: vader включає в себе широкий список слів, фраз та емодзі, які часто використовуються в соціальних медіа, дозволяючи йому точно оцінювати сентимент в таких текстах;
- правилобазований підхід: vader використовує набір заздалегідь визначених лінгвістичних правил для аналізу сентименту, що дозволяє йому швидко та ефективно обробляти текст без потреби в тренуванні моделей на великих даних;
- миттєві результати: завдяки своїй правилобазованій природі, vader може швидко аналізувати великі обсяги тексту без додаткових обчислювальних витрат, що ідеально підходить для середовищ, де швидкість відгуку є критичною.

Далі првоедемо експеремент на реальних датасетах які будуть відображати різні характеристики неструктурованих текстових відгуків.

На рисунку 3.4 зображено три речення які ми будемо оцінювати, далі створимо список із 3 речень, у кожному з яких буде одне позитивне, одне негативне та одне нейтральне речення. І тому ми можемо побачити, як працюють VADER і TextBlob на кожному з них. Відповідно на рисунку 3.5 зображено результат відпрацювання бібліотек.

З рисунку вище ми можемо зробити висновок, що VADER ідеально визначив перше речення як нейтральне речення, де TextBlob не такий вже й далекий від нього. Тоді для другого речення VADER дає позитивну оцінку, але TextBlob дає нам більш позитивну оцінку. І для останнього речення VADER дає більш негативний бал, ніж TextBlob.

На рисунку 3.5 неведемо текст у якому додано роздільні знаки. На рисунку 3.6 неведено результат.

```
text_list = ["This is my first ever post on the internet.",
             "I am very excited to write this post.",
             "It's not good to work late hours."]
```

Рисунок 3.5 – Приклад з роздільними знаками

```
sentence: This is my first ever post on the internet.
 VADER sentiment score: 0.0
 TextBlob score: 0.25
==============================
sentence: I am very excited to write this post.
 VADER sentiment score: 0.4005
 TextBlob score: 0.48750000000000004
==============================
sentence: It's not good to work late hours.
 VADER sentiment score: -0.3412
 TextBlob score: -0.32499999999999996
==============================
```

Рисунок 3.6 – Результат тексту з роздільними знаками

Тепер, з наведеної вище комірки, ми можемо сказати, що знак оклику справді покращує нашу оцінку в усіх реченнях. Але для нашого нейтрального речення (речення № 1) TextBlob відривається.

На рисунку 3.7 неведемо текст у якому додано капіталізацію. На рисунку 3.8 неведено результат.

```
text_list = ["This is my first ever post on the internet!",
             "I am very excited to write this post!",
             "It's not good to work late hours!"]
```

Рисунок 3.7 – Приклад з капіталізацією

```
sentence: This is my first ever post on the internet!
 VADER sentiment score: 0.0
 TextBlob score: 0.3125
==============================
sentence: I am very excited to write this post!
 VADER sentiment score: 0.4561
 TextBlob score: 0.609375
==============================
sentence: It's not good to work late hours!
 VADER sentiment score: -0.4015
 TextBlob score: -0.3625
==============================
```

Рисунок 3.8 – Результат тексту з капіталізацією

Тепер ми можемо сказати, що наша оцінка VADER покращилася, але оцінка TextBlob залишилася незмінною. Причина в тому, що VADER вважає, що версія з великими літерами має сильніший настрій і збільшила оцінку настрою. У той же час TextBlob не розрізняв настрої між верхнім і нижнім регістром слова.

На рисунку 3.9 неведемо текст у якому додано емодзі. На рисунку 3.10 неведено результат.

```
text_list = ["This is my FIRST EVER post on the internet!",
             "I am very EXCITED to write this post!",
             "It's NOT GOOD to work late hours!"]
```

Рисунок 3.9 – Приклад з імодзі

```
sentence: This is my FIRST EVER post on the internet!
 VADER sentiment score: 0.0
 TextBlob score: 0.3125
==============================
sentence: I am very EXCITED to write this post!
 VADER sentiment score: 0.5744
 TextBlob score: 0.609375
==============================
sentence: It's NOT GOOD to work late hours!
 VADER sentiment score: -0.5007
 TextBlob score: -0.3625
==============================
```

Рисунок 3.10 – Результат тексту з імодзі

Ми чітко бачимо, що показник VADER покращується. Але оцінка TextBlob зовсім не змінюється.

Хоча в дослідженні структурованих відгуків філіпінських громадян у одному із соціальних опитувань показало кращі результати для TextBlob[(див. рис. 3.1.1).

Evaluation

TextBlob Evaluation Matrix:

	precision	recall	f1-score	support
negative	1.00	1.00	1.00	7
neutral	1.00	1.00	1.00	18
positive	1.00	1.00	1.00	75
accuracy			1.00	100
macro avg	1.00	1.00	1.00	100
weighted avg	1.00	1.00	1.00	100

VADER Evaluation Matrix:

	precision	recall	f1-score	support
negative	1.00	0.29	0.44	7
neutral	0.38	0.17	0.23	18
positive	0.80	0.96	0.87	75
accuracy			0.77	100
macro avg	0.72	0.47	0.52	100
weighted avg	0.74	0.77	0.73	100

Рисунок 3.11 – Опитування стандартизованих та офіційних відгуків

Дивлячись на TextBlob, він показує ідеальні оцінки за всіма показниками. Це говорить про те, що алгоритму TextBlob вдалося правильно класифікувати всі настрої в наборі відгуків про Філіпіни.

4 РОЗРОБКА ОРГАНІЧНОГО ПОЕДНАННЯ ДВОХ ПІДХОДІВ

Поєднання функціоналу та можливостей бібліотек VADER і TextBlob для аналізу сентименту може бути дуже корисним, оскільки вони компенсують одна одну слабкості та розширюють спектр аналітичних можливостей.

VADER ефективний в розпізнаванні сентименту в неформальних текстах з соціальних медіа завдяки своїй здатності до аналізу емодзі та сленгу, тоді як TextBlob використовує методи машинного навчання, які можуть бути більш адаптивними до різноманітних джерел тексту та надають можливість до-навчання на специфічних датасетах[12].

Це поєднання дозволяє ефективно обробляти широкий спектр текстових даних, забезпечуючи більш точне та глибоке розуміння сентименту. Інтеграція цих бібліотек може допомогти уникнути помилок, що виникають через контекстуальні нюанси мови, і забезпечити більш універсальне рішення для аналізу сентименту, що може бути використане в різних аналітичних завданнях від моніторингу бренду до аналізу споживчих відгуків.

4.1 Розробка стратегії поєднання

Розробка бібліотеки, яка б просто збирала результати аналізу сентименту від двох різних бібліотек, таких як VADER і TextBlob, і обчислювала їх середнє значення, може здатися зручною на перший погляд, але на практиці це не завжди найкращий підхід з кількох причин.

По-перше, VADER і TextBlob використовують різні методології для оцінки сентименту, що може призвести до значної варіативності в їх оцінках для одних і тих же текстів. VADER більш ефективний для коротких, емоційно насичених текстів, які часто зустрічаються в соціальних медіа, включаючи емодзі та сленг, тоді як TextBlob може краще справлятися з більш формалізованими текстами та більш точно обробляти більш складні синтаксичні структури завдяки своїм алгоритмам машинного навчання. В результаті, просте середнє значення двох оцінок може не відображати реальний сентимент тексту, оскільки кожна бібліотека може інтерпретувати текст по-різному.

По-друге, просте усереднення може призвести до "розмивання" більш крайніх або виразних оцінок. Наприклад, якщо одна бібліотека дуже позитивно оцінює текст, а інша – дуже негативно, середнє значення може вийти нейтральним, що є помилковим представленням обох вихідних оцінок.

Крім того, є питання ефективності такого підходу. Використання середнього значення як метрики для сентименту може не враховувати нюанси або особливості окремих випадків, де більш складні статистичні методи, такі як медіана, можуть бути більш інформативними. Медіана, на відміну від середнього значення, є більш стійкою до викидів і може надати краще уявлення про "центральну тенденцію" розподілу оцінок сентименту, що може бути корисним у випадках, коли дані мають високу ступінь розкиду або асиметрію.

Таким чином, підхід до обчислення середнього значення сентименту між двома бібліотеками без урахування контексту та методологічних різниць може не бути найефективнішим способом аналізу. Це може призвести до втрати важливої інформації про сентимент, яка б могла бути корисною при прийнятті рішень на основі текстових даних.

4.2 Ключова ідея

Ідея та стратегія полягає в гармонійному поєднанні цих підходів, підход наступний - спочтку ми зберемо дані з однакового тексту та проженем по двох цих бібліотеках, після цього ми створемо з двох результатів - дві колекції ключ-значення де значенням буде оцінка кожного слова кожною відповідною бібліотекою, після цього ми пройдемось по кожному слову(в обох колекціях) порівнюючи одні й ті самі елементи та відберем такі - різниця між якими - достатньо велика(30%), так само ми зробимо із комбінаціями слів (комбінація токенів або фраз), після цього оскільки ми вже з'ясували що Vader краще з'ясовує емоційну забарвленість - ми замінимо в фінальній спільній коллекції(яку зробимо шляхом поєднання двох де перерахуємо ключі всим словам) такі слова(комбінація токенів або фраз) із TextBlob на слова із VADER, а всі інші слова де різниця менше 30% - візьмемо середне між двома колекціями.

Наш підхід до поєднання результатів аналізу сентименту від VADER та TextBlob досить інноваційний та зосереджений на усуненні відмінностей у враженнях сентименту між цими двома бібліотеками. Такий підхід може допомогти максимально використати сильні сторони кожної бібліотеки, забезпечуючи більш точне та надійне розуміння емоційного забарвлення тексту.

Ми плануємо відбирати слова та фрази, де різниця в оцінках між двома бібліотеками є значною (30% та більше), і віддавати перевагу оцінкам від VADER для виразів з сильним емоційним зарядом. Це важливо, оскільки VADER, як ви вже зазначили, має кращу здатність розпізнавати емоційне забарвлення, особливо в текстах соціальних медіа, що включають емодзі, сленг та інші специфічні вирази.

Для словників, де різниця в оцінках менша за 30%, ви плануєте використовувати середнє значення, що дозволить забезпечити баланс між різними підходами до аналізу сентименту. Це може допомогти згладити будь-які надмірні відхилення, які можуть виникнути, якщо одна бібліотека дає неточну оцінку. На рисунку 4.1 схематично зображено процес поєднання.

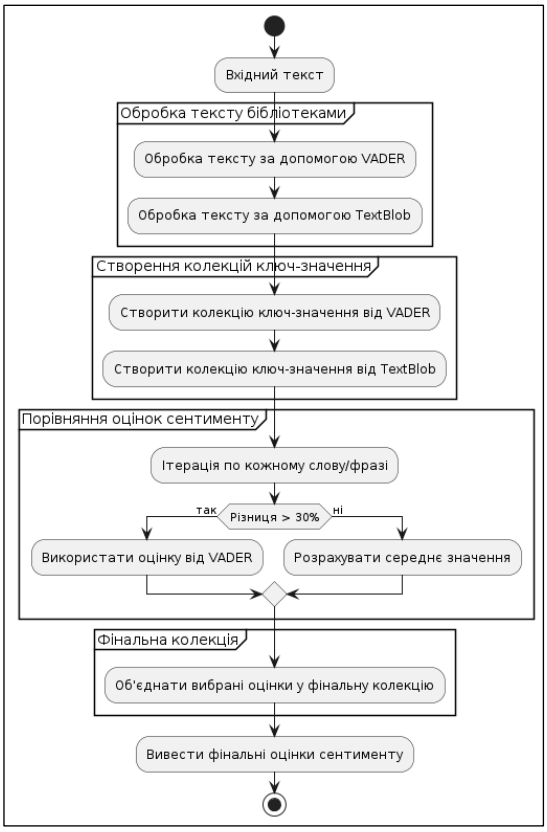

Рисунок 4.1.1 – Схематичний процес поєднання

4.3 Розробка системи класів

На рисунку 4.2 наведено схему системи класів для застосунку по реалізації методу поєднання.

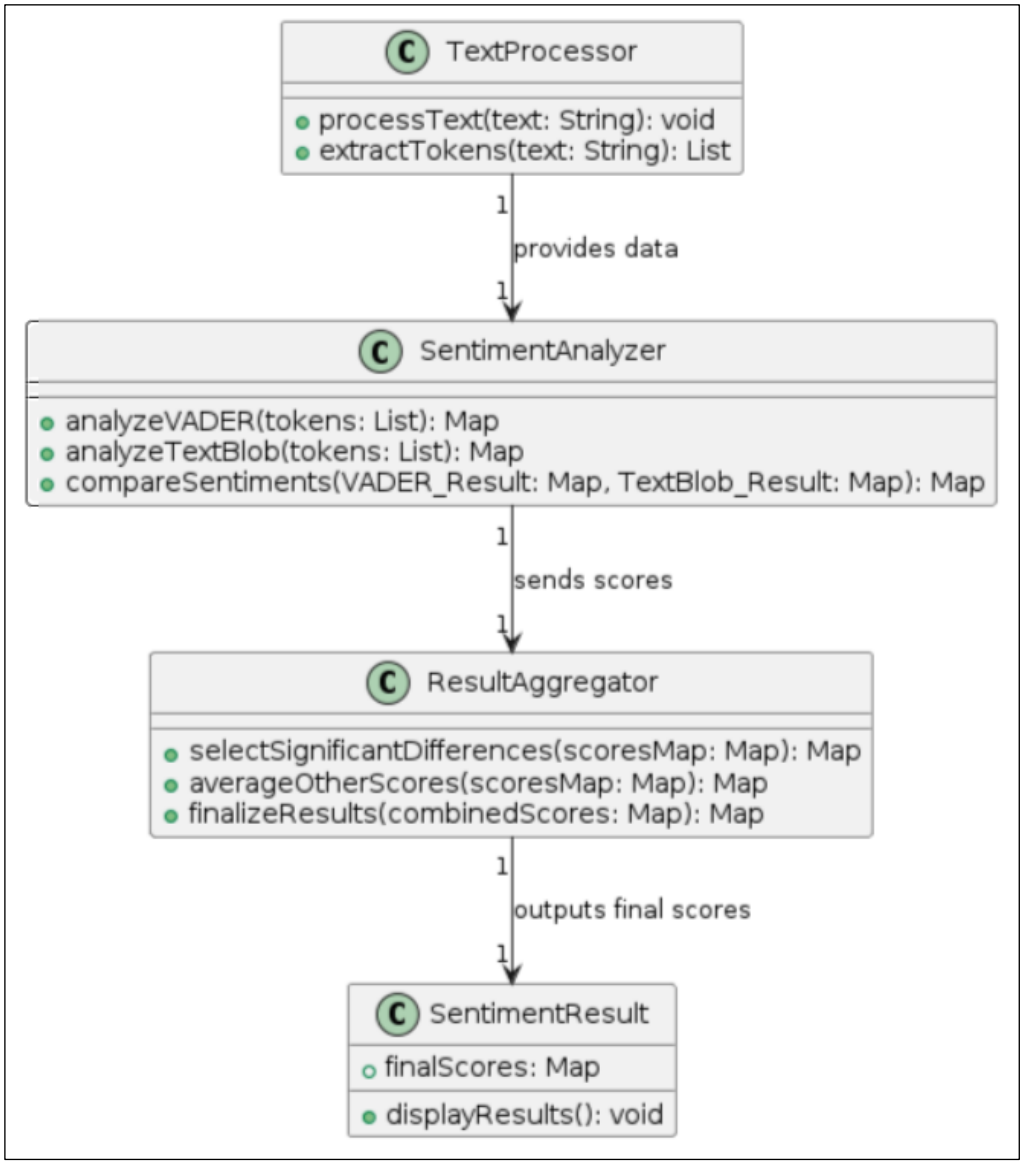

Рисунок 4.2 – Схему системи класів

Опис компонентів системи:

– textprocessor: відповідає за первинну обробку тексту. цей клас бере сирий текст, процесує його для подальшого аналізу (наприклад, токенізація);
– processtext: метод, що приймає текст та виконує його підготовку;
– extracttokens: розділяє текст на токени, які потім аналізуються;

- sentimentanalyzer: здійснює аналіз сентименту за допомогою двох бібліотек (vader і textblob);
- analyzevader та analyzetextblob: обидва методи беруть токени від textprocessor та повертають мапу (словник) сентиментних оцінок;
- comparesentiments: порівнює результати від двох бібліотек і визначає, чи відмінності в оцінках є значущими;
- resultaggregator – відповідає за визначення кінцевого сентименту, об'єднуючи результати;
- selectsignificantdifferences: вибирає значення з більшою різницею в оцінках;
- averageotherscores: обчислює середнє значення для слов/фраз, де різниця в оцінках менша за порогове значення;
- finalizeresults: комбінує всі оцінки в кінцевий набір результатів.
- sentimentresult: цей клас зберігає та управляє кінцевими результатами аналізу сентименту;
- displayresults: метод для відображення або виведення кінцевих оцінок сентименту.

Ця система класів спрямована для гнучкого та ефективного аналізу сентименту, де кожен компонент виконує свою специфічну задачу, але разом вони формують інтегровану систему аналізу сентименту.

Нижче наведемо код основного класу поєднання. Рисунок наведено у додатку Б.

```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
from textblob import TextBlob
import nltk
nltk.download('punkt')

class TextProcessor:
    def tokenize_text(self, text):
        return nltk.word_tokenize(text)

class SentimentAnalyzer:
    def __init__(self):
        self.vader_analyzer = SentimentIntensityAnalyzer()
```

```
    def analyze_with_vader(self, tokens):
        scores = {}
        for token in tokens:
            score = self.vader_analyzer.polarity_scores(token)['compound']
            scores[token] = score
        return scores

    def analyze_with_textblob(self, tokens):
        scores = {}
        for token in tokens:
            score = TextBlob(token).sentiment.polarity
            # Normalize TextBlob scores to match VADER's scale
            normalized_score = (score + 1) / 2
            scores[token] = normalized_score
        return scores

class ResultAggregator:
    def combine_results(self, vader_scores, textblob_scores):
        final_scores = {}
        for token in vader_scores:
            if abs(vader_scores[token] - textblob_scores.get(token, 0)) >
0.3:
                final_scores[token] = vader_scores[token] if
vader_scores[token] > textblob_scores.get(token, 0) else
textblob_scores[token]
            else:
                final_scores[token] = (vader_scores[token] +
textblob_scores.get(token, 0)) / 2
        return final_scores

class SentimentAnalysisSystem:
    def __init__(self, text):
        self.text = text
        self.processor = TextProcessor()
        self.analyzer = SentimentAnalyzer()
        self.aggregator = ResultAggregator()

    def perform_analysis(self):
        tokens = self.processor.tokenize_text(self.text)
        vader_scores = self.analyzer.analyze_with_vader(tokens)
        textblob_scores = self.analyzer.analyze_with_textblob(tokens)
        final_scores = self.aggregator.combine_results(vader_scores,
textblob_scores)
        return final_scores

# Example usage
text = "I love this product! It works amazingly well. Best purchase ever!!!
Not happy with the service though."
system = SentimentAnalysisSystem(text)
results = system.perform_analysis()
print(results)
```

Опис коду:

– textprocessor: токенізує текст для подальшого аналізу;

- sentimentanalyzer: аналізує токени, використовуючи vader та textblob, і повертає словник з оцінками сентименту;
- resultaggregator: об'єднує результати від двох аналізаторів. якщо різниця оцінок більше 30%, вибирається більша оцінка. в інших випадках обраховується середнє;
- sentimentanalysissystem: керує процесом аналізу від початку до кінця, від токенізації тексту до отримання кінцевих оцінок сентименту.

ВИСНОВКИ

У даній роботі було проведено дослідження аналізу сентименту на основі неструктурованих текстових відгуків, за допомогою популярних бібліотек аналізу сентименту: VADER і TextBlob. Головною метою роботи було розробити та валідувати модель, що ефективно комбінує результати цих двох підходів для підвищення точності виявлення сентименту у текстах.

Аналіз показав, що інтеграція результатів обох бібліотек дозволяє досягти вищої точності порівняно з використанням кожної бібліотеки окремо. Зокрема, було виявлено, що VADER ефективніший для виявлення емоційної забарвленості текстів, що містять іронію та сленг, тоді як TextBlob показує кращі результати у стандартних висловлюваннях. Поєднуючи їхні сильні сторони, можна адекватно обробляти ширший спектр текстових форматів.

Практичне застосування розробленої моделі може бути надзвичайно корисним у сферах, де важливе швидке та точне розуміння сентименту користувачів – від маркетингу та соціальних медіа аналізів до підтримки клієнтів та виробничих систем.

У майбутньому планується розширення дослідження з використанням глибших нейронних мереж для аналізу змін у сентименті протягом часу та розробка методів автоматичного підбору ваг між результатами двох бібліотек в залежності від контексту використання. Також важливим аспектом є подальше тестування моделі на більшій та різноманітній вибірці текстів для забезпечення її універсальності та надійності.

Цей підхід до аналізу сентименту показав, що використання комплексних інструментів може значно підвищити якість та ефективність обробки текстових даних, що відкриває нові перспективи для розвитку технологій обробки природної мови.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Bing Liu. Sentiment Analysis: Mining Opinions, Sentiments, and Emotions., URL: (дата звернення 10.04.2024);

2. Duyu Tang, Bing Qin, Ting Liu. Deep Learning for Sentiment Analysis: A Survey. , URL: (дата звернення 10.04.2024);

3. Manning, C. D., & Schütze, H. (1999). Foundations of Statistical Natural Language Processing. MIT Press, URL: (дата звернення 10.04.2024);

4. Jurafsky, D., & Martin, J. H. (2020). Speech and Language Processing (3rd ed.). Pearson, URL: (дата звернення 10.04.2024);

5. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space, URL: (дата звернення 10.04.2024);

6. Вихідців Е.І. Математичні моделі й методи рішення завдань прогнозування рівнів забруднення прикордонного шару атмосфери сильнодіючими отруйними речовинами при аварійних викидах // Праці 4-го Міжнародного молодіжного форуму «Радіоелектроніка й молодь у XXI столітті». -Частина 2. - Харків: ХГТУРЭ. -2000.- С.6-7. (дата звернення 18.04.2024);

7. Клименко Е.Г. Програмно-алгоритмічні засоби інтелектуального аналізу даних // Радіоелектроніка й інформатика. - 2001. - № 3. - С. 64-67. (дата звернення 18.04.2024);

8. K. Guntupally, R. Devarakonda and K. Kehoe, "Spring Boot based REST API to Improve Data Quality Report Generation for Big Scientific Data: ARM Data Center Example," 2018 IEEE International Conference on Big Data (Big Data), 2018, pp. 5328-5329, doi: 10.1109/BigData.2018.8621924. (дата звернення 18.04.2024);

9. Aggarwal S. Modern web-development using reactjs //International Journal of Recent Research Aspects. – 2018. – Т. 5. – №. 1. – С. 2349-7688., URL: (дата звернення 18.04.2024);

10. Models of adaptive integration of weighted interval data in tasks of predictive expert assessment / I. Ruban et al. Eastern-European Journal of Enterprise

Technologies. 2022. Vol. 5, no. 4(119). P. 6–15., URL: https://doi.org/10.15587/1729-4061.2022.265782 (дата звернення 19.04.2024);

11. Hacimahmud A. V., Khakhanova H., Litvinova E. Vector Logic Analysis of Big Data. 2023 IEEE East-West Design & Test Symposium (EWDTS), Batumi, Georgia, 22–25 September 2023. 2023. URL: https://doi.org/10.1109/ewdts59469.2023.10297032 (дата звернення 19.04.2024).

12. Evaluation and Analysis of the NLP Model Zoo for Ukrainian Text Classification / D. Panchenko et al. Information and Communication Technologies in Education, Research, and Industrial Applications. Cham, 2022. P. 109–123. URL: https://doi.org/10.1007/978-3-031-20834-8_6 (дата звернення 19.04.2024).