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 оцінки сентименту діалогових повідомлень
(тема)

Виконав:
студент (ка) 2 курсу, групи ІПЗм-22-5

Штанько О.О.
(прізвище, ініціали)

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

Керівник доцент кафедри ПІ, к.т.н., 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 Штаньку Олексію Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження методів оцінки сентименту діалогових повідомлень»

Затверджена наказом по університету від 29.13. 2024р. № 250 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 20.06.2024

3. Вихідні дані до роботи вимоги до обробки набору даних для проведення досліджень за обраною предметною областю, мови програмування C#, технології .NET 8.0, Telegram API, середовища розробки Visual Studio 2022

4. Перелік питань, що потрібно опрацювати в роботі
аналіз та порівняння існуючих методів оцінки текстів за емоційним забарвленням, вибір підходящих методів для дослідження, проектування логічної моделі даних для проведення експериментальних досліджень, написання програмних рішень, проведення експериментів та аналіз отриманих результатів

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Видача завдання	30.03.2024	виконано
2	Аналіз предметної галузі	12.04.2024	виконано
3	Постановка задачі	12.04.2024	виконано
4	Експериментальні дослідження	12.04.2024 – 20.04.24	виконано
5	Аналіз результатів експериментальних досліджень та розробка рекомендацій	20.04 – 28.05.24	виконано
6	Написання та оформлення статті та тез доповіді	28.05 – 10.06.24	виконано
7	Підготовка пояснювальної записки	10.06 – 11.06.24	виконано
8	Підготовка презентації та доповіді	12. 06.24	виконано
9	Нормоконтроль	13.06.24	виконано
10	Рецензування	14.06.24	виконано
11	Занесення диплома в електронний архів	15.06.2024	виконано
12	Попередній захист	17.06.2024	виконано
13	Допуск до захисту у зав. кафедри	19.06.2024	виконано

Дата видачі завдання «30» березня 2024 р.

Студент _____ Штанько О. О.

(підпис)

Керівник роботи_____ доцент Турута О. П.

(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка містить 61 ст., 28 рис., 1 табл., 15 джерел.

АНАЛІЗ СЕНТИМЕНТУ, ДІАЛОГОВІ ПОВІДОМЛЕННЯ, ОБРОБКА ПРИРОДНОЇ МОВИ, ЕМОЦІЙНИЙ АНАЛІЗ ТЕКСТУ, МЕТОДИ ОЦІНКИ СЕНТИМЕНТУ, ЕФЕКТИВНІСТЬ АЛГОРИТМІВ, КОРПУС ТЕКСТІВ, МАШИННЕ НАВЧАННЯ.

Об'єктом дослідження є методи оцінки сентименту в діалогових повідомленнях. Дослідження спрямоване на вивчення та порівняння різних підходів, використовуваних для визначення емоційного витягу текстів, які обмінюють користувачі в месенджерах та інших комунікаційних платформах.

Метою дослідження є визначення ефективності та придатності різних методів оцінки сентименту в діалогових повідомленнях. Основний акцент робиться на вивченні можливостей застосування цих методів у реальних умовах та їхньому порівнянні з точки зору точності та широкого спектру застосувань.

Результати дослідження полягатимуть у визначенні найбільш ефективних методів оцінки сентименту в діалогових повідомленнях, а також у розкритті їхнього потенціалу та можливостей в реальних умовах. Дані результати можуть бути використані для подальшого розвитку систем аналізу текстових повідомлень та впровадження їх в різноманітні галузі, де важливий аналіз сентименту.

ALGORITHM EFFICIENCY, DIALOG MESSAGES, EMOTIONAL TEXT ANALYSIS, MACHINE LEARNING, NATURAL LANGUAGE PROCESSING, SENTIMENT ANALYSIS, SENTIMENT ASSESSMENT METHODS, TEXT CORPUS.

The object of the study is methods of sentiment evaluation in dialogue messages. The study is aimed at studying and comparing different approaches used to determine

the emotional content of texts exchanged by users in messengers and other communication platforms.

The purpose of the study is to determine the effectiveness and suitability of various methods of sentiment evaluation in dialogue messages. The main emphasis is on studying the possibilities of applying these methods in real conditions and comparing them in terms of accuracy and a wide range of applications.

The results of the research will consist in determining the most effective methods of sentiment evaluation in dialogue messages, as well as in revealing their potential and opportunities in real conditions. These results can be used for the further development of text message analysis systems and their implementation in various fields where sentiment analysis is important.

Я, Штанько Олексій Олександрович, студент гр. ІПЗм-22-5, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження методів оцінки сентименту діалогових повідомлень», 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 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

В сучасному цифровому суспільстві, в контексті активного використання діалогових повідомлень на різних комунікаційних платформах, виникає велика необхідність в розвитку ефективних методів аналізу сентименту в цифрових текстових діалогах. Спілкування через месенджери, соціальні мережі та інші онлайн-платформи стає нелишенню частиною нашого повсякденного життя, а отже, розуміння емоційного витягу таких повідомлень набуває великої вагомості.

Актуальність дослідження полягає у тому, що емоційний контекст текстових повідомлень може мати значущий вплив на різноманітні аспекти, починаючи від взаємодії між користувачами та закінчуючи аналізом громадської думки та трендів. Аналіз сентименту в діалогових повідомленнях може бути корисним для бізнесу, політики, соціальних досліджень та інших галузей, де важливо розуміти емоційний настрій спільноти.

Мета даної курсової роботи полягає в проведенні дослідження та порівнянні різних методів оцінки сентименту в діалогових повідомленнях. В рамках роботи буде розглянуто різні підходи, від традиційних методів машинного навчання до сучасних технік обробки природної мови. Основним завданням є визначення ефективності та застосовності цих методів у реальних умовах.

Дослідження може послужити підґрунтям для подальших розвитків у галузі аналізу емоційного витягу текстових повідомлень та впровадження його у різноманітні сфери сучасного суспільства.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Аналіз емоційного відтінку тексту

Аналіз тональності тексту, також відомий як сентимент-аналіз або opinion mining, представляє собою групу методів у сфері комп'ютерної лінгвістики, призначених для автоматизованого виявлення в текстах емоційно забарвленої лексики та оцінки автором емоційного ставлення до об'єктів, що розглядаються в тексті [1].

Термін "тональність" вказує на відношення автора висловлювання до теми, що розглядається в тексті, будь то об'єкт реального світу, подія, процес чи їх властивості. Лексична тональність або лексичний сентимент виражається на рівні окремих слів або комунікативних фрагментів. Тональність всього тексту можна визначити як функцію, наприклад, суму лексичних тональностей його компонентів (речень) і правил їх комбінування.

Аналіз тональності текстів є сферою обробки природної мови (Natural Language Processing, NLP), яка спрямована на розробку систем, що намагаються ідентифікувати та витягти висловлені думки у тексті. Окрім ідентифікації думок, ці системи також визначають атрибути висловленого, такі як:

- полярність: вказує, чи виражено оратором позитивну чи негативну думку;
- тема: вказує на те, про що саме йдеться у тексті;
- власник думки: визначає особу чи організацію, яка висловлює свою думку.

На сьогодні аналіз настроїв стає динамічно розвиваючим напрямом, оскільки має безліч практичних застосувань. З великим обсягом доступної інформації в Інтернеті на різних платформах, таких як рецензійні сайти, форуми, блоги і соціальні медіа, системи аналізу настроїв можуть автоматично перетворювати цю неструктуровану інформацію в структуровані дані, що відображають громадську думку щодо продуктів, послуг, брендів, політики та інших тем. Ці дані стають цінним ресурсом для комерційних застосувань, таких

як маркетинговий аналіз, взаємодія з громадськістю, огляди продуктів, визначення чистого промоутера, аналіз відгуків клієнтів та інші області.

Текстову інформацію можна розділити на два основні типи: факти та думки. Факти є об'єктивними висловами про щось, тоді як думки, як правило, представляють суб'єктивні вирази, що описують почуття, оцінки та емоції людей щодо конкретної теми чи об'єкта.

Аналіз сентиментів, подібно до багатьох інших задач обробки природної мови (NLP), може бути визначений як проблема класифікації, де основним завданням є розв'язання двох підпроблем.

Класифікація речення як суб'єктивне чи об'єктивне: це відомо як класифікація суб'єктності.

Класифікація речення як вираз позитивної, негативної чи нейтральної думки: це відомо як класифікація полярності.

Думка, виражена в тексті, може стосуватися об'єкта, його складових, аспектів, атрибутів або особливостей. Об'єктом може бути продукт, послуга, особа, організація, подія чи тема. Наприклад: "Термін служби акумулятора цієї камери надто короткий" виражає негативну думку про особливість (час автономної роботи) суб'єкта (камери).

Існують два типи думок: прямі і порівняльні. Прямі висновки висловлюють думку безпосередньо, наприклад: "Якість зображення камери А погана." – це пряма негативна думка про камеру А.

У порівняльних висновках думка виражається за допомогою порівняння об'єкта, що розглядається в тексті, з іншим, часто подібним об'єктом, наприклад: "Якість зображення камери А перевершує якість камери В." Ці порівняльні думки, як правило, вказують на подібності або відмінності між двома чи більше об'єктами, використовуючи порівняльні або вищі форми прикметників чи прислівників. У зазначеному прикладі виражена позитивна думка про камеру А і, навпаки, негативна думка про камеру В.

Явна думка на тему – це думка, яку виразно висловлюється в суб'єктивному реченні. Наприклад: "Якість звуку цього телефону вражає своєю відмінністю."

Імпліцитна думка є висловленою в об'єктивному реченні, але може підтримувати неявну думку. Наприклад: "Наушник зламався через два дні". У неявних думках можуть містити метафори, які представляють собою складний тип висловлення думок, оскільки вони надають багато семантичної інформації.

1.2 Типи аналізу тональності тексту

У сучасних системах автоматичного визначення емоційної оцінки тексту часто використовується простий, одномірний емотивний простір, який виражається у двох основних вимірах: позитивний або негативний (добре або погано). Однак існують випадки успішного використання багатовимірних просторів.

Основне завдання в аналізі тональності полягає в класифікації полярності тексту, тобто визначенні, чи виражена в тексті думка позитивно, негативно чи нейтрально. Розширена класифікація тональності може включати емоційні стани, такі як "злий", "сумний" і "щасливий" [2].

Полярність документа може бути визначена за бінарною шкалою, де два класи оцінок використовуються для визначення полярності документа: позитивна чи негативна.

Однак недоліком цього підходу є те, що емоційну складову документа не завжди можна однозначно визначити, тобто документ може містити як ознаки позитивної, так і негативної оцінки. Ранні дослідження в цій області включають роботи Терні та Панга, які використовують різні методи розпізнавання полярності для оглядів товарів та відгуків на фільми відповідно. Це є прикладом роботи на рівні документа.

Полярність документа може бути класифікована на багатосмуговій шкалі, що було виконано Пангом і Снайдером (серед інших). Вони розширили основне завдання класифікації кіновідгуків, перейшовши від оцінки «позитивний чи негативний» до прогнозування рейтингу за 3- або 4-бальною шкалою. Снайдер також провів глибокий аналіз оглядів ресторанів, передбачаючи рейтинги різних характеристик, таких як їжа і атмосфера, за 5-бальною шкалою.

Інший метод визначення тональності включає в себе використання системи шкалювання, де слова, зазвичай пов'язані з негативними, нейтральними або позитивними тональностями, призначаються відповідні числа від -10 до 10 (від негативного до дуже позитивного). Спочатку фрагмент неструктурованого тексту піддається аналізу за допомогою інструментів та алгоритмів обробки природної мови, після чого виділені об'єкти та терміни аналізуються для розуміння значення цих слів.

Ще одним напрямком досліджень є визначення суб'єктивності/об'єктивності в тексті. Зазвичай це завдання формулюється як віднесення даного тексту до одного з двох класів: суб'єктивний або об'єктивний. Ця проблема іноді може бути складнішою, ніж класифікація полярності, оскільки суб'єктивність слів і фраз може залежати від контексту, і об'єктивний документ може містити в собі суб'єктивні пропозиції, наприклад, у новинах, які цитують думки людей. Крім того, як зазначав Су, результати в значній мірі залежать від визначення суб'єктивності, що використовується при анотуванні текстів. Тим не менше, Панг продемонстрував, що видалення об'єктивних пропозицій з документа перед класифікацією полярності сприяє підвищенню точності результатів.

Ще одна модель, яка використовується для більш детального аналізу, називається аналізом на основі функції/аспекту. Ця модель відноситься до ухвалення думок або настроїв, виражених різними функціями або аспектами об'єктів, такими як мобільний телефон, цифрова камера або банк. Функція/аспект представляє собою атрибут або компонент сутності, який досліджується на наявність тональності, наприклад, екран мобільного телефону або якість фотозйомки камери. Ця проблема передбачає вирішення ряду завдань, таких як ідентифікація актуальних сутностей, витяг їх функцій та аспектів, а також визначення того, чи є думка, висловлена щодо кожної функції/аспекту, позитивною, негативною або нейтральною. Докладніше про це можна знайти в довіднику з NLP у розділі "Аналіз тональності та суб'єктивності".

Існує різноманіття підходів до аналізу настроїв, і інструменти визначення тональності варіюються від систем, які фокусуються на полярності (позитивні,

негативні, нейтральні), до систем, які враховують почуття та емоції (сердиті, щасливі, сумні і т.д.), або визначають наміри (наприклад, зацікавлені проти не зацікавлених). У наступному розділі ми розглянемо основні підходи до цього завдання.

Іноді можна бути більш точним у визначенні рівня полярності думки, застосовуючи категорії, такі як:

- дуже позитивний;
- позитивний;
- нейтральний;
- негативний;
- дуже негативний.

Цей підхід часто називають дрібнозернистим аналізом. Наприклад, його можна використовувати при оцінці відгуків на основі 5-зіркового рейтингу, де "дуже позитивний" відповідає 5 зіркам, а "дуже негативний" - 1 зірці.

Деякі системи також розрізняють різні відтінки полярності, визначаючи, чи асоціюються позитивні або негативні настрої з певними почуттями, такими як гнів, смуток або турбота (негативні почуття), або щастя, любов чи ентузіазм (позитивні почуття).

Виявлення емоцій спрямоване на розпізнавання різних почуттів, таких як щастя, розчарування, гнів, смуток та інші. Часто системи виявлення емоцій використовують лексикони (списки слів та їх емоційних відтінків) або складні алгоритми машинного навчання.

1.3 Методи аналізу тональності тексту

Існує кілька методів аналізу тональності тексту, які використовуються для визначення емоційної забарвленості та вираження думок у текстових документах [3]. Ось деякі з них:

- аналіз лексики (лексичний аналіз);
- машинне навчання (machine learning);
- правила та евристики;

– векторне представлення слів (word embeddings);

– глибоке навчання (deep learning);

– комбіновані методи.

Кожен метод має свої переваги та обмеження, і вибір конкретного підходу може залежати від конкретних вимог завдання та характеристик текстових даних.

1.3.1 Аналіз лексики

Цей метод використовує словники або лексикони, що містять слова та їх емоційні забарвлення. Кожне слово призначається певній емоційній вагі, і тональність тексту визначається за сумою цих ваг. Однак цей підхід може бути обмежений врахуванням контексту та синонімів.

Аналіз лексики або лексичний аналіз в контексті аналізу тональності ґрунтується на використанні словників або лексиконів, що містять емоційні забарвлення слів. Кожне слово або фраза з тексту порівнюється зі словами в цьому лексиконі, і визначається його емоційна оцінка (позитивна, негативна або нейтральна).

Кроки лексичного аналізу.

Підготовка лексикону: створення словника, в якому кожне слово має присвоєний емоційний показник. Цей словник може бути розроблений вручну експертами або використовувати готові лексикони.

Визначення слова: кожне слово чи фраза з аналізованого тексту розглядається окремо.

Порівняння з лексиконом: слово порівнюється зі словами в лексиконі, і йому присвоюється емоційна оцінка згідно з його значенням в лексиконі.

Агрегація оцінок: емоційні оцінки для кожного слова об'єднуються для визначення загальної емоційної оцінки тексту. Зазвичай використовуються підсумкові метрики, такі як загальна кількість позитивних слів мінус кількість негативних.

Переваги лексичного аналізу.

Простота і швидкість: лексичний аналіз є відносно простим і швидким методом, особливо, якщо використовується готовий лексикон.

Інтерпретованість: результати лексичного аналізу легко інтерпретовані, оскільки вони базуються на конкретних словах та їх емоційних оцінках.

Недоліки лексичного аналізу.

Врахування контексту: метод не завжди враховує контекст, у якому використовується слово, що може призводити до неточностей.

Специфічність мови: деякі вирази мають різне емоційне забарвлення в різних контекстах або для різних груп людей.

Проблема зі словами-омонімами: слова, які мають кілька значень, можуть викликати плутанину.

Лексичний аналіз часто використовується в простих випадках аналізу тональності, де контекст не важливий, і важливо просто визначити загальний настрій тексту. Цей метод може бути ефективним для експрес-оцінки великої кількості текстів.

1.3.2 Машинне навчання

Машинне навчання (МН) в аналізі тональності тексту передбачає навчання моделей на великих корпусах текстових даних, щоб вони могли автоматично визначати емоційні оцінки для нових текстів. Це може включати в себе класифікацію текстів за позитивними, негативними або нейтральними емоційними відтінками.

Кроки використання машинного навчання для аналізу тональності.

Підготовка даних: формування навчального набору даних, що включає в себе текстові приклади разом з емоційними мітками (позитивний, негативний, нейтральний).

Вибір моделі: вибір типу моделі машинного навчання. Найчастіше використовуються класифікатори, такі як наївний байєсівський класифікатор, метод опорних векторів (SVM), рішення дерева, глибокі нейронні мережі тощо.

Навчання моделі: подача навчальних даних в обрану модель для навчання. Модель виробляє внутрішні параметри на основі поданих даних.

Валідація моделі: перевірка ефективності моделі на валідаційному наборі даних для перевірки її здатності правильно класифікувати нові тексти.

Тестування: перевірка роботи моделі на тестовому наборі даних для оцінки загальної ефективності та уникнення перенавчання.

Переваги машинного навчання.

Контекстуальна чутливість: машинні навчальні моделі можуть враховувати контекст та взаємодію слів в реченнях, що дозволяє їм ефективно розрізняти відтінки емоцій.

Адаптабельність: моделі можуть адаптуватися до нових даних, що дозволяє їм покращувати точність з часом.

Недоліки машинного навчання.

Потреба в великих наборах даних: машинні навчальні моделі часто потребують великих обсягів даних для ефективного навчання.

Складність моделей: деякі моделі, особливо глибокі нейронні мережі, можуть бути складними для розуміння та налаштування.

Машинне навчання ефективно використовується в сучасних системах аналізу тональності для розрізнення складних контекстів і ефективного класифікування текстів за їхнім емоційним забарвленням. Воно забезпечує адаптивність і точність в порівнянні з традиційними методами.

1.3.3 Правила та евристики

Поза методами машинного навчання, існують правила та евристики, які можуть застосовуватися для аналізу тональності тексту. Ці підходи базуються на лінгвістичних та семантичних особливостях мови і не вимагають великих обсягів даних для навчання.

Приклади правил та евристик.

Аналіз семантичної структури.

Збалансованість висловлювань: визначення кількості позитивних та негативних словосполучень у тексті.

Інтенсивність слів: визначення емоційно забарвлених слів та їхньої інтенсивності.

Врахування контексту.

Обробка подвійних знаків: перевірка вживання подвійних знаків пунктуації (наприклад, "чудово!!!" або "погано???"), що може вказувати на інтенсивність емоцій.

Виявлення синонімів та антонімів.

Використання антонімів: врахування вживання слів з протилежним емоційним забарвленням.

Обробка негацій.

Виявлення негаційних слів: врахування вживання слів, які змінюють сенс на протилежний (наприклад, "не погано").

Врахування лексичного контексту.

Зв'язок слів у реченні: оцінка емоційного забарвлення на основі синтаксичних зв'язків між словами у реченні.

Визначення іронії.

Обробка іронічних висловлювань: розпізнавання слів та висловів, які можуть вказувати на іронію чи сарказм.

Врахування емоційних виразів.

Аналіз емотиконів та emoji: врахування емоційних знаків, які часто використовуються в текстах для вираження настрою.

Ці правила і евристики можуть використовуватися окремо чи в поєднанні з іншими методами, дозволяючи проводити аналіз тональності на різних рівнях складності та структурності текстів.

1.3.4 Векторне представлення слів

Векторне представлення слів – це техніка, що дозволяє представляти слова у вигляді числових векторів у просторі так, щоб схожі слова були близькими за

значенням у цьому просторі. Це надає можливість моделям машинного навчання працювати зі словами як з числами та взаємодіяти з ними на основі семантичних відносин.

Методи векторного представлення слів.

Однорідні вектори (One-Hot Vectors).

Кожне слово представляється вектором, де всі елементи, окрім одного, рівні нулю. Це "одиничне" значення вказує на позначене слово.

Word2Vec.

Модель, яка навчається генерувати вектори слів на основі їхнього контексту в тексті. Зберігає семантичні відносини між словами.

GloVe (Global Vectors).

Модель, яка використовує статистику глобальних залежностей між словами для створення їхніх векторних представлень.

FastText.

Розширення Word2Vec, яке враховує морфологічні особливості слів, розбиваючи їх на підрядки.

BERT (Bidirectional Encoder Representations from Transformers).

Модель, яка використовує контекстне векторне представлення слов на основі навчання на великих корпусах тексту.

Використання в аналізі тональності.

Класифікація.

Векторне представлення слів дозволяє використовувати слова як вхід для класифікаторів машинного навчання для визначення емоційної тональності тексту.

Кластеризація.

Схожі вектори можуть вказувати на схожі за значенням слова, що полегшує кластеризацію слів за емоційним забарвленням.

Аналіз схожості.

Можна вимірювати схожість між векторами слів, що полегшує визначення слов, які можуть мати схожі емоційні відтінки.

Векторне представлення слів є потужним інструментом в аналізі тональності, оскільки воно враховує контекст та семантичні зв'язки між словами, сприяючи кращому розумінню емоційного забарвлення тексту.

1.3.5 Глибоке навчання

Глибоке навчання – це галузь машинного навчання, яка використовує нейронні мережі з глибокою структурою (глибокі нейронні мережі) для вирішення завдань навчання та прогнозування. Основною ідеєю є використання багатошарових архітектур для представлення високорівневих функцій та здатності витягати складні взаємозв'язки в даних. Глибоке навчання добре справляється з завданнями, де важлива абстракція та вищий рівень представлення даних.

Основні характеристики глибокого навчання.

Багатошарові нейронні мережі.

Глибоке навчання використовує багатошарові нейронні мережі, такі як згорткові нейронні мережі (CNN) та рекурентні нейронні мережі (RNN), які дозволяють представляти дуже складні залежності у вхідних даних.

Автоматичне витягування функцій.

Мережі глибокого навчання можуть автоматично витягувати важливі функції з даних без необхідності ручної інженерії ознак.

Здатність до роботи з великими обсягами даних.

Глибоке навчання найбільш ефективне при навчанні на великих обсягах даних, що робить його відмінним варіантом для завдань, де великі набори даних є ключовими.

Універсальність.

Здатність глибокого навчання пристосовуватися до різноманітних завдань, від класифікації та регресії до генерації контенту та обробки природної мови.

Застосування глибокого навчання в аналізі тональності.

Аналіз сентиментів.

Глибоке навчання використовується для класифікації текстів на позитивні, негативні або нейтральні сентименти.

Генерація контенту.

Моделі глибокого навчання можуть генерувати текст із заданими емоційними відтінками, використовуючи векторне представлення слів.

Відеоаналітика.

Використання глибокого навчання для аналізу емоцій та сентименту у відео контенті.

Глибинний аналіз зображень.

Застосування глибоких мереж для аналізу виразів обличчя та визначення емоцій.

Глибоке навчання є потужним інструментом в аналізі тональності, оскільки воно дозволяє автоматично визначати складні емоційні зв'язки та витягати значущі функції з текстового та візуального контенту.

1.3.6 Комбіновані методи

Комбіновані методи в аналізі тональності об'єднують різні підходи та техніки для отримання більш точних та надійних результатів. Це може включати в себе комбінацію правилових підходів, машинного навчання та лінгвістичних методів. Нижче представлені деякі комбіновані методи в аналізі тональності:

Правила та машинне навчання.

Використання правилових підходів для обробки структурованих частин тексту (наприклад, визначення сутностей) та використання машинного навчання для аналізу неструктурованих частин.

Комбінація аспект-орієнтованого та сентиментного аналізу.

Аналізування емоцій та настроїв, що стосуються конкретних аспектів сутностей. Наприклад, визначення, які аспекти продукту викликають позитивні або негативні відгуки.

Лінгвістичні та векторні методи.

Використання лінгвістичних методів для аналізу семантичної структури тексту та векторних представлень слів для розуміння контексту.

Комбіновані підходи до виявлення емоцій.

Об'єднання акустичного аналізу (наприклад, емоційного голосу) з аналізом тексту для отримання повнішого розуміння сентименту.

Комбіновані методи фільтрації шуму.

Використання методів фільтрації шуму для видалення надлишкової інформації та покращення точності аналізу.

Метафоричний аналіз і виокремлення контексту.

Врахування метафоричних висловлювань та аналіз контексту для зменшення можливих спотворень в значенні тексту.

Динамічні та адаптивні моделі.

Розробка моделей, які можуть динамічно адаптуватися до змін в мові та виразі емоцій з часом.

Комбіновані методи дають змогу використовувати переваги різних підходів і компенсувати їхні недоліки, що сприяє покращенню якості аналізу тональності. Такі підходи можуть бути особливо ефективними в умовах складних і варіативних джерел тексту та емоцій.

1.4 Постановка задачі

Дослідження ставить перед собою дві ключові задачі. У першу чергу, вона націлюється на вивчення та розробку методу розпізнавання іменованих сутностей в неструктурованих текстах, визначаючи імена, прізвища та по батькові. Для досягнення цієї мети планується:

- огляд існуючих методів розпізнавання іменованих сутностей, спрямованих як на іноземні мови, так і на українську мову;
- розробка програмної реалізації розпізнавання іменованих сутностей;
- створення корпусу даних, що включатиме в себе неструктуровані тексти українською мовою;
- проведення тестування та оцінка ефективності розробленого методу.

Друга значуща задача роботи полягає в дослідженні та розробці методу сентиментального аналізу текстів українською мовою. Для досягнення успіху в цьому напрямку планується:

- аналіз існуючих методів аналізу сентименту в текстах, орієнтованих як на іноземні мови, так і на українську мову;
- реалізація програмного забезпечення для методу сентиментального аналізу тексту;
- створення корпусу даних, що включатиме в себе неструктуровані тексти українською мовою;
- тестування та оцінка якості використаних методів аналізу сентименту.

Обидві задачі спрямовані на вирішення актуальних завдань у галузі обробки природної мови та роботи з неструктурованим текстом, дозволяючи отримати нові уявлення про ефективність використовуваних методів на українських текстових джерелах.

2 МЕТОДИ ВИРІШЕННЯ ПРОБЛЕМИ

2.1 Аналіз метрик

Існує різноманіття методів для отримання показників ефективності та оцінки точності класифікатора при аналізі настроїв, але одним із найбільш популярних і широко використовуваних є перехресна перевірка.

Принцип дії перехресної перевірки полягає в розбитті тренувальних даних на дві частини: одну для навчання класифікатора (зазвичай 75% від усіх даних) та іншу для його тестування (залишаючи 25% для тестів). Модель навчається на тренувальних даних і перевіряється на тестових складах, вимірюючи її продуктивність. Цей процес повторюється декілька разів, і середнє значення обчислюється для кожної метрики.

Уникаючи постійного використання одного і того ж тестового набору, який може призвести до перенавчання (overfitting) для конкретного набору даних, перехресна перевірка допомагає запобігти цьому. З більшою кількістю даних можна ефективно використовувати різні тестові набори.

Оцінка продуктивності класифікатора часто базується на таких стандартних показниках, як точність, продуктивність та час відклику. Точність визначає, яка частина текстів була правильно визначена як належна до певної категорії відносно всіх передбачених текстів. При оцінці ефективності важливо також враховувати, що зі збільшенням обсягу даних для навчання класифікатора його точність може значно покращитися.

Найчастіше для вимірювання продуктивності класифікатора в аналізі настроїв використовують точність та час відклику. Точність є ключовим показником, оскільки вона визначає, яка частина текстів була правильно визначена як належна до певної категорії відносно всіх передбачених текстів. Однак точність сама по собі може бути недостатньою для повного розуміння ефективності класифікатора.

У складних завданнях, таких як аналіз настроїв, точність та час відклику, ймовірно, будуть на низькому рівні, особливо на початкових етапах, коли

класифікатор має обмежену інформацію для навчання. Проте з подачею класифікатору більше даних його продуктивність може значно покращитися.

При оцінці угоди між анотаторами, що визначає, наскільки однаково різні анотатори маркували тексти для аналізу настроїв, часто використовується Криппендорфська альфа. Цей показник дає уявлення про те, наскільки згодні різні анотатори в роботі з анотаціями. Наприклад, значення 0.655 для Криппендорфської альфи у дослідженні аналізу настроїв в Twitter свідчить про значну, хоча не ідеальну, угоду між анотаторами.

Важливо враховувати, що угода між анотаторами може бути важливою у випадках, коли анотовані дані не є абсолютно точними, і визначення анотаційного стандарту впливає на надійність отриманих результатів.

Популярні метрики оцінки ефективності методів оцінки сентименту в діалогових повідомленнях:

– точність (Accuracy): це відношення правильно класифікованих повідомлень до загальної кількості повідомлень. Формула 2.1:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{2.1}$$

де TP (True Positives): Кількість правильно класифікованих позитивних елементів;

TN (True Negatives): Кількість правильно класифікованих негативних елементів;

FP (False Positives): Кількість неправильно класифікованих позитивних елементів;

FN (False Negatives): Кількість неправильно класифікованих негативних елементів.

– точність (Precision): це відношення правильно класифікованих позитивних екземплярів до загальної кількості позитивно класифікованих екземплярів. Формула 2.2:

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \quad (2.2)$$

– полнота (Recall): це відношення правильно класифікованих позитивних екземплярів до загальної кількості позитивних екземплярів. Формула 2.3:

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + FN} \quad (2.3)$$

– F-мера (F1 Score): це гармонічний середній Precision і Recall. Формула 2.4:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (2.4)$$

– матриця плутанини (Confusion Matrix): це таблиця, яка показує кількість True Positives, True Negatives, False Positives і False Negatives. З цієї матриці можна вивести багато інших метрик;
– ROC-крива і площа під ROC-кривою (ROC-AUC): ROC-крива відображає відношення True Positive Rate (Recall) до False Positive Rate при різних порогах вирішення. Площа під ROC-кривою вимірює загальну ефективність класифікатора.

Ці метрики можуть бути корисними в залежності від конкретних завдань і вимог дослідження сентименту в діалогових повідомленнях.

2.2 Дослідження датасетів

Основною складовою успішного вивчення аналізу настроїв є робота з різноманітними наборами даних та експериментування з різними підходами. По-перше, необхідно здобути дані та створити набір, над яким будуть проводитися експерименти, враховуючи власний домен та інтереси.

Нижче перераховано деякі з найпопулярніших наборів даних для експериментів з аналізу настроїв та використання методів машинного навчання. Вони є відкритими та доступними для безкоштовного завантаження[5].

Відгуки про продукт.

Мільйони відгуків клієнтів Amazon із зірковими рейтингами, дуже корисні для тренування моделей аналізу настроїв.

Відгуки ресторанів.

5,2 мільйонів відгуків Yelp із рейтингами зірок.

Огляди фільмів.

1000 позитивних і 1000 негативних оброблених відгуків, а також 5,331 позитивних і 5,331 негативних оброблених фрагментів.

Точні відгуки про продукти харчування.

Приблизно 500 000 відгуків про продукти харчування від Amazon із інформацією про продукт і користувача, рейтингами та текстовими версіями кожного огляду.

Відгуки авіакомпанії Twitter на Kaggle.

Приблизно 15 000 позначених твітів (позитивних, нейтральних і негативних) про авіакомпанії.

Перший дебат у групі GOP Twitter Sentiment.

Приблизно 14 000 позначених твітів (позитивних, нейтральних та негативних) щодо першої дискусії GOP у 2016 році.

Якщо ви цікавитесь підходами, заснованими на правилах, нижче представлений різноманітний список лексиконів аналізу настроїв, які можуть бути корисні. Ці лексикони містять словники з мітками, що визначають їхні настрої в різних контекстах.

Ці лексикони є дійсно корисними для визначення настрою текстів і можуть служити важливим інструментом для аналізу емоційного відтінку мовлення. Ось короткий опис кожного з них:

Лексикони настроїв для 81 Мови.

Включає позитивні та негативні лексики настроїв для 81 мови, що дозволяє проводити аналіз текстів різних мов.

SentiWordNet.

Містить близько 29 000 слів із оцінкою настроїв від 0 до 1, де 0 вказує на негативний настрій, 1 - на позитивний, а значення між ними представляє ступінь настрою.

Лексикон думки для аналізу настроїв.

Складається із списку 4782 негативних та 20000 позитивних слів англійською мовою, що допомагає ідентифікувати настрій у текстах.

Слово словника Wordstat.

Містить приблизно 4800 позитивних і 9000 негативних слів, що використовуються для визначення настрою у текстах.

Emoticon Sentiment Lexicon.

Зберігає список 477 смайликів, класифікованих як позитивні, нейтральні або негативні, що допомагає розпізнавати емоційний відтінок.

Лексикон AFINN.

Одна з найпростіших і популярних лексиконів для аналізу настроїв. Містить більше трьох тисяч слів із відомою оцінкою полярності. Доступний у вільному доступі на GitHub.

Ці інструменти можуть бути ефективними для використання в аналізі настроїв у текстах та розвитку моделей емоційного аналізу.

2.3 API для оцінки емоційного стану текстів

Існує кілька альтернатив систем аналізу настроїв, які доступні через API. Загалом їх можна розподілити на дві основні категорії: відкриті бібліотеки та комерційні рішення [6].

Мова програмування .NET є однією з провідних у галузі розробки програмного забезпечення, має потужну спільноту та великий набір інструментів для реалізації моделей обробки природної мови (NLP).

Бібліотека ML.NET пропонує інструменти для машинного навчання та векторизації тексту, таких як tf-idf або векторизатори тексту, і її легко використовувати для навчання класифікаторів.

NLTK.NET – адаптація традиційної бібліотеки NLTK для .NET, яка має активну спільноту та, крім великого набору функцій для обробки природної мови, надає можливість тренування класифікаторів машинного навчання.

SpaCy.NET – ще одна нова бібліотека NLP для .NET із зростаючою спільнотою. Подібно до NLTK.NET, вона пропонує потужний набір низькорівневих функцій для обробки природної мови та підтримку навчання класифікаторів тексту. За останні кілька років завдяки розвитку глибокого навчання виникла нова група комп'ютерних бібліотек, які сприяють застосуванню обробки природної мови (NLP).

TensorFlow.NET, розроблений компанією Google, надає набір інструментів для створення та навчання нейронних мереж. Він також підтримує векторизацію тексту, включаючи традиційну частоту слова та більш складні вбудовування слів.

Keras.NET відображає корисні абстракції для роботи з різними типами нейронних мереж, такими як рекурентні нейронні мережі (RNNs) і згорткові нейронні мережі (CNNs), і легко взаємодіє з TensorFlow.NET або Theano.NET. Він також забезпечує інструменти для класифікації тексту. TorchSharp, підтримуваний такими організаціями, як Facebook, Twitter, Nvidia та іншими, є останньою системою глибокого навчання для .NET зі сильною громадою.

TextBlob.NET, інша відкрита бібліотека, дозволяє легко виконувати завдання NLP, включаючи аналіз настроїв. Вона використовує лексикон настроїв для оцінки полярності та суб'єктивності текстів, де оцінки знаходяться в нормалізованому масштабі.

2.4 Дослідження обраних методів аналізу

2.4.1 Мережі нейронів зі згортанням

Згорткові нейронні мережі (CNN) - це клас моделей глибокого навчання, що зазвичай використовуються для обробки зображень, відео та природних мов.

Вони є регуляризованою версією багатошарових персептронів і розроблені таким чином, щоб потребувати мінімальної попередньої обробки даних для роботи [7].

Робота згорткових нейронних мереж зазвичай полягає в перетворенні конкретних даних в абстрактніші представлення, доки не досягається вищий рівень абстракції. Мережа автоматично виробляє ієрархію абстрактних ознак чи послідовностей ознак, фільтруючи неважливі деталі та виділяючи важливі.

Одержані в результаті роботи нейронної мережі ознаки можуть бути складними для розуміння. Якщо система пропускає важливі ознаки, рекомендується вдосконалити структуру та архітектуру мережі замість зміни змісту ознак.

У звичайному перцептроні, який є повнозв'язною нейронною мережею, кожен нейрон пов'язаний з усіма нейронами попереднього шару, при цьому кожне з'єднання має власний ваговий коефіцієнт [7].

У згортковій нейронній мережі в операції згортки використовується обмежена матриця ваг невеликого розміру, яку «пересувають» по всьому оброблюваному шару (спочатку - безпосередньо за вхідними даними), де вона формує сигнал активації для нейрона наступного шару після кожного зсуву з аналогічною позицією (див. рис. 2.1).

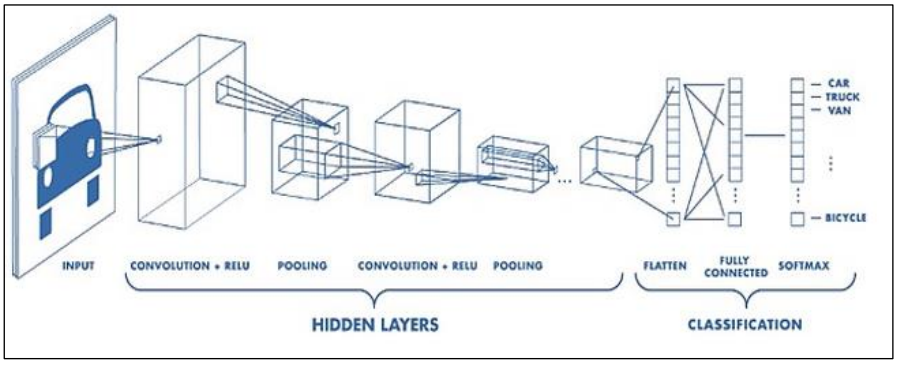


Рисунок 2.1 – Архітектура згорткової нейронної мережі

Отже, для різних нейронів вихідного шару використовується та сама матриця ваг, яку також називають ядром згортки. Її можна трактувати як графічне кодування певного ознаки, наприклад, наявність похилої лінії під певним кутом чи певних фігур, що повторюються на різних зображеннях. Тоді наступний шар,

який виникає в результаті операції згортки з цією матрицею ваг, показує наявність цієї ознаки в оброблюваному шарі та її координати, утворюючи так звану карту ознак (англ. Feature map). При цьому такі ядра згортки не передбачаються відразу, а завжди формуються самостійно за допомогою навчання мережі класичним методом зворотного розповсюдження помилки.

Згорткові нейронні мережі використовують не лише один набір ваг, а цілу гаму, яка визначає елементи вхідних даних, такі як лінії та дуги під різними кутами на зображенні. Ці ядра згортки не задаються наперед, а формуються під час навчання мережі за допомогою методу зворотного поширення помилки. Кожен набір ваг створює свою власну карту ознак, що робить нейронну мережу багатоканальною (з багатьма незалежними картами ознак на одному шарі). При переміщенні по шару матрицею ваг її зсувають не на весь крок, а на невелику відстань, щоб уникнути пропускання шуканих ознак. Наприклад, при розмірі матриці ваг 5 × 5 її зсувають на один або два нейрона замість п'яти, щоб не пропустити необхідну ознаку.

Один із найпростіших і популярних методів навчання - це метод навчання з учителем, який ґрунтується на маркованих даних, зокрема метод зворотного поширення помилки та його варіації. Проте існують також методи навчання без учителя для згорткових мереж. Наприклад, фільтри згорткових операцій можна навчати незалежно, подаючи на них випадкові вирізки з вихідних зображень навчальної вибірки та застосовуючи до них різноманітні алгоритми навчання без учителя, такі як автоассоціатори або метод k-середніх. Цей підхід відомий як patch-based training. В такий спосіб наступний шар згорткової мережі навчатиметься на вирізках, отриманих від вже навченого першого шару. Також можна поєднувати згорткові нейронні мережі з іншими технологіями глибинного навчання, наприклад, створювати згорткові автоассоціатори, згорткові версії каскадних обмежених машин Больцмана, які навчаються за допомогою імовірнісного математичного апарату, або згорткові версії розрідженого кодування, відомі як розгорткові мережі [8].

Для аналізу функціонування згорткових нейронних мереж планується застосування бібліотеки Keras. Ця відкрита платформа надає засоби для маніпулювання вхідними даними так, щоб вони були придатні для використання в мережі.

2.4.2 Рекурентна нейронна мережа – довга та короткочасна пам’ять

Рекурентна нейронна мережа (RNN) – тип нейронних мереж, де вузли з’єднані у напрямлену у часі послідовність (орієнтований граф) [9], дозволяючи обробляти послідовні просторові ланцюги або серії подій у часі. Завдяки цій особливості рекурентні нейронні мережі можуть працювати з послідовностями будь-якої довжини, використовуючи свою внутрішню пам'ять. Це робить їх популярними для завдань розпізнавання тексту або мови, а також для обробки природних мов [10], де вони можуть використовувати здобутий досвід.

Найпоширеніші конфігурації архітектури рекурентних нейронних мереж включають моделі з довгою короткочасною пам'яттю (LSTM) та керовані рекурентні блоки (GRU).

У дослідженні буде використана архітектура з довгою короткочасною пам'яттю.

LSTM-мережі ефективно використовуються для класифікаційних задач [11]. Ці нейронні мережі містять LSTM-модулі як основу або доповнення до інших модулів.

LSTM-модуль – це рекурентний модуль, що здатний запам'ятовувати значення на короткі та довгі інтервали часу. Використання LSTM-модулів не призводить до зникнення градієнта під час навчання мережі завдяки відсутності функції активації всередині рекурентних модулів і стійкому збереженню значень у часі.

LSTM-модулі часто об'єднують у блоки, що є характерною особливістю глибоких багатошарових нейронних мереж, що дозволяє використовувати паралельні обчислення та спеціальне обладнання.

LSTM-блоки включають три або чотири "вентиля" (gates), які регулюють потік інформації на входах та виходах пам'яті цих блоків (рис. 2.2).

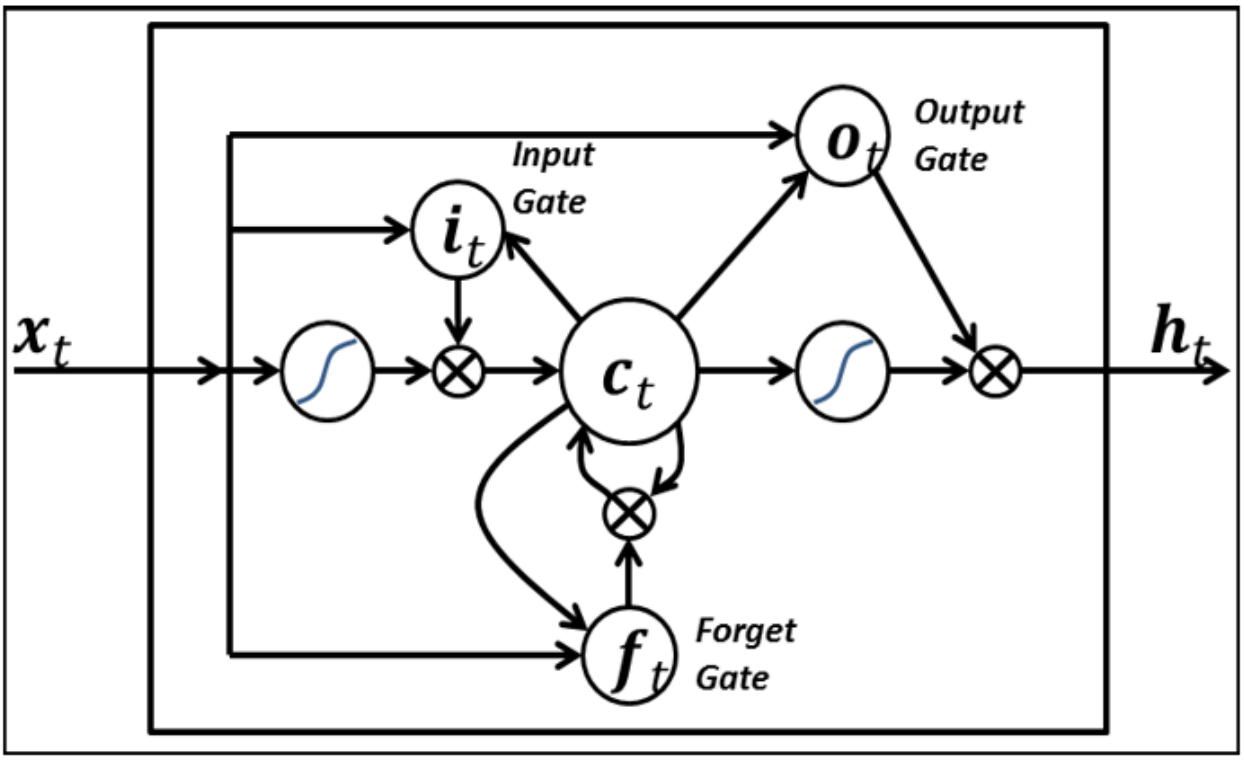


Рисунок 2.2 – Блок LSTM з трьома вентилями

Вентилі у вигляді логістичних функцій використовуються для регулювання потоку інформації в середину та назовні пам'яті, де кожен з них має свою функцію: "вхідний вентиль" визначає, чи має нове значення ввійти до пам'яті, "вентиль забуття" контролює збереження значень у пам'яті, а "вихідний вентиль" регулює використання значень для активації виходу блоку [12]. Ваги (W і U) визначаються для керування цими вентилями на основі вхідних даних та попереднього виходу блоку. Тренування цих ваг дозволяє LSTM-блоку оптимізувати свою функцію шляхом мінімізації втрат. LSTM-блоки зазвичай навчаються за допомогою методу зворотного поширення помилки в часі.

Лінійні послідовності часто використовуються для різних завдань, таких як керування роботами, розпізнавання мовлення, навчання граматики, визначення дій людей, прогнозування часових рядів, навчання ритму та виявлення гомології білків [13].

2.4.3 Попередньо навчені моделі нейронних мереж

Модель, попередньо підготовлена, це збережена мережа, яка раніше навчалася на великому наборі даних, зазвичай, на широкому завданні класифікації зображень. У такому підході використовується або модель, яка вже була оброблена, або використовується трансферне навчання для того, щоб адаптувати цю модель до конкретного завдання.

Трансферне навчання – це метод машинного навчання, при якому модель, навчена на одному завданні, переноситься на інше відповідне завдання. Згідно з визначенням у книзі "Deep Learning" [14], трансферне навчання та адаптація домену - це ситуації, коли знання, набуті в одній ситуації, використовуються для поліпшення узагальнення в іншій. Це покращення навчання в новому завданні через передачу знань з вже вивченого пов'язаного завдання.

Ідея трансферного навчання полягає у тому, що якщо модель навчена на широкому та різноманітному наборі даних, то вона може ефективно служити загальною моделлю візуального світу. Це означає, що отримані знання можна використовувати без необхідності починати навчання з нуля на великому обсязі даних.

Останні роки принесли значний прогрес у глибокому навчанні, дозволяючи вирішувати складні задачі і досягати вражаючих результатів. Однак час і обсяг даних, необхідних для таких систем, є значно вищими, ніж для традиційних методів машинного навчання. На сьогоднішній день існують різноманітні глибокі нейронні мережі з передовими можливостями (іноді навіть кращими, ніж у людини), які були розроблені та протестовані у різних областях, таких як комп'ютерне зорове сприйняття та обробка природної мови [15]. Більшість команд або дослідників діляться деталями цих мереж, щоб інші могли скористатися їхніми досягненнями. Ці передньо навчені мережі або моделі стають основою трансферного навчання в глибокому навчанні, відомого як "глибоке трансферне навчання".

Для аналізу настроїв у тексті можна використовувати універсальні моделі обробки природної мови. Ці моделі знаходять застосування у різних областях,

таких як машинний переклад, системи відповідей на запитання, чат-боти та інші. Основною складовою цих універсальних моделей є концепція мовного моделювання.

Найвідоміші попередньо натреновані моделі включають:

- ULMFiT: ця модель навчена на даних з Wikitext, що дозволяє їй виконувати широкий спектр завдань обробки природної мови;
- Transformer: ця модель, представлена Google, була навчена за допомогою згорткових та рекурентних нейронних мереж;
- Google’s BERT: інша попередньо натренована модель від Google, яка включає результати 11 завдань обробки природної мови, зокрема аналіз тональності тексту;
- ELMo: ця модель використовує вбудовування слів для перетворення текстових даних у числові.

У даному дослідженні буде використана модель Google’s BERT, яка представляє собою метод попередньої підготовки мови. Це означає, що модель "розуміння мови" тренується на великому корпусі тексту, такому як Вікіпедія, а потім використовується для різних завдань обробки природної мови. BERT виходить за межі попередніх методів, оскільки він є першою безнаглядною, глибоко двонапрямленою системою попередньої підготовки мови для обробки природної мови .

У цьому контексті "Без нагляду означає", що BERT був навчений лише на загальних текстових даних, що є значущим, оскільки велика кількість таких даних доступна в Інтернеті на багатьох мовах. Попередньо навчені моделі можуть мати контекстно-вільні або контекстуальні представлення, а контекстні представлення можуть бути однонаправленими або двонаправленими. Наприклад, моделі, такі як word2vec або GloVe, створюють однакові "вбудовання" для кожного слова у словнику, тобто слово "банк" матиме однакове представлення у контексті банківського депозиту та річкового берега. У контекстних моделях представлення кожного слова формується на основі інших слів у реченні.

3 ПРОВЕДЕННЯ ДОСЛІДЖЕННЯ

3.1 Інструменти та дані для дослідження

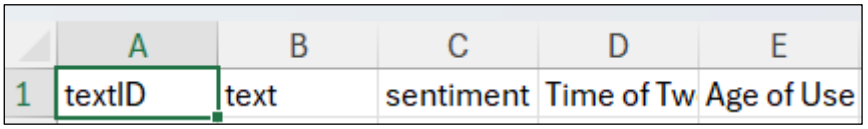
Для проведення аналізу був використаний набір даних, що включає 75 000 відгуків на фільми з бази IMDB. Кожен відгук у цьому наборі даних має мітку "позитивний" або "негативний", що дозволяє виконувати бінарний аналіз тональності текстів. Всі відгуки написані англійською мовою. Наприклад, ось позитивний відгук до фільму "Титанік":

"Why do people criticize this film while ignoring truly awful ones like The Godfather? Titanic stands as the pinnacle of 21st-century cinema, boasting exceptional acting, direction, effects, music, and overall quality. Yet, it's consistently disparaged simply because someone voiced a negative opinion, leading the majority to follow suit. There's nothing inherently flawed about Titanic; in fact, it's the most decorated film in Oscar history and the highest-grossing film ever. Regrettably, it remains one of the most underrated movies I've encountered. In truth, it's unrivaled as the greatest film of all time, surpassing even the likes of Star Wars, The Lord of the Rings trilogy, or works by masters like Hitchcock, Spielberg, or Tim Burton. While these are undoubtedly commendable films and directors, none can match the brilliance of James Cameron's masterpiece, Titanic."

Одним із прикладів негативних відгуків про фільм "Титанік" є такий:

"First viewed on May 17, 2002 - Rating: 3 out of 10 (Director: Ewald Andre Dupont): This early adaptation of the Titanic tragedy falls short, offering a lackluster portrayal of the historic event. While the depiction of the disaster itself is passable, the surrounding drama often feels contrived, with poor acting and an overly melodramatic tone. The narrative closely mirrors that of the more recent, Oscar-winning film, albeit with a focus on the crew's attempts to conceal the severity of the situation until it's almost too late. Recommended solely for those seeking a nostalgic glimpse into past interpretations of the Titanic story and to appreciate the advancements made by James Cameron's acclaimed rendition."

Відгуки у наборі для навчання (75,000 відгуків) структуровані та містять однакову кількість позитивних та негативних відгуків (див. рис. 3.1). Крім того, набір даних має ще 25,000 відгуків без інформації про тон тексту.



	A	B	C	D	E
1	textID	text	sentiment	Time of Tw	Age of Use

Рисунок 3.1 – Структура набору даних

Було обрано платформу .NET для проведення дослідження в галузі машинного навчання. .NET є потужною кросплатформовою екосистемою, розробленою компанією Microsoft і вперше випущеною в 2002 році. Її архітектура та об'єктно-орієнтований підхід спрямовані на допомогу розробникам у написанні чіткого та логічного коду для проектів будь-якого масштабу.

У сучасному світі .NET широко використовується для виконання завдань у галузі машинного навчання, наукових обчислень, веб-розробки, настільних та мобільних додатків, а також хмарних сервісів. Простота використання та підтримка багатьох мов програмування, таких як C#, F#, і Visual Basic, роблять .NET дуже привабливою для проведення досліджень та розробки у сфері машинного навчання. Однією з найбільших переваг .NET є велика кількість бібліотек та інструментів, які можна використовувати для наукових досліджень, зокрема у сфері машинного навчання та нейронних мереж.

Однією з основних бібліотек для машинного навчання в .NET є ML.NET. Ця бібліотека дозволяє створювати, навчати та впроваджувати моделі машинного навчання безпосередньо в .NET додатках. ML.NET підтримує широкий спектр алгоритмів, включаючи класифікацію, регресію, кластеризацію та інші. Крім того, .NET підтримує інтеграцію з іншими популярними інструментами та бібліотеками для машинного навчання, такими як TensorFlow та ONNX.

Це дозволяє використовувати переваги сучасних технологій машинного навчання в межах .NET екосистеми. Нижче буде описано використані в дослідженні бібліотеки та платформи.

Для роботи з згортковими нейронними мережами буде використано TensorFlow. Це відкрита платформа з відкритим вихідним кодом для машинного навчання, яка має широку та гнучку екосистему інструментів, бібліотек та ресурсів спільноти. Це дозволяє дослідникам використовувати сучасні технології розробки машинного навчання, а розробникам легко створювати та розгортати додатки, які працюють на основі машинного навчання.

Для інтеграції моделей машинного навчання в .NET-додатки буде використовуватися ML.NET. ML.NET – це кросплатформова бібліотека з відкритим кодом для машинного навчання, розроблена компанією Microsoft. Вона дозволяє створювати, навчати та впроваджувати моделі машинного навчання безпосередньо в .NET-додатках. ML.NET підтримує широкий спектр алгоритмів, включаючи класифікацію, регресію, кластеризацію та інші.

Для взаємодії з TensorFlow у межах .NET-додатків буде використовуватися ONNX (Open Neural Network Exchange). ONNX – це відкрита екосистема для взаємодії моделей машинного навчання між різними фреймворками. Вона дозволяє розробникам використовувати переваги різних інструментів та бібліотек для створення та навчання моделей, а потім експортувати їх у формат ONNX для подальшого використання в інших середовищах.

Нижче буде описано використані в дослідженні бібліотеки та платформи: TensorFlow:

Широко використовується для створення моделей глибокого навчання, включаючи згорткові нейронні мережі.

Забезпечує високу продуктивність і масштабованість.

Підтримує апаратні прискорювачі, такі як GPU та TPU.

ML.NET.

Кросплатформова бібліотека для машинного навчання, розроблена компанією Microsoft.

Підтримує широкий спектр алгоритмів машинного навчання.

Дозволяє створювати, навчати та впроваджувати моделі машинного навчання безпосередньо в .NET-додатках.

ONNX.

Відкрита екосистема для взаємодії моделей машинного навчання між різними фреймворками.

Дозволяє експортувати моделі з TensorFlow та інших фреймворків у формат ONNX для подальшого використання в інших середовищах.

Використання цих бібліотек та платформ дозволяє поєднати переваги сучасних технологій машинного навчання з потужністю та гнучкістю .NET екосистеми, що забезпечує ефективний процес розробки та впровадження складних моделей машинного навчання.

Для створення користувацького інтерфейсу використовується Telegram API. Це потужний інструмент з відкритим вихідним кодом для розробки ботів та інтеграції з месенджером Telegram. Telegram API дозволяє розробникам створювати різноманітні додатки та сервіси, які можуть взаємодіяти з користувачами через інтерфейс Telegram.

Основним середовищем розробки обрано Visual Studio від Microsoft, що є інтегрованим середовищем розробки (IDE) для платформи .NET. Visual Studio надає потужні інструменти для аналізу коду, графічний відладчик, інструменти для запуску юніт-тестів та підтримує розробку веб-додатків, мобільних додатків, хмарних сервісів та багато іншого.

Платформою для цього використовується Windows.

3.2 Оцінка ефективності використання згорткових нейронних мереж

ML.NET вміє використовувати набір даних відгуків з IMDB для тренування моделі. Формат цього набору даних готовий для використання в нейронних мережах, тому не потрібно ні переводити текст до нижнього регістру, ні видаляти знаки пунктуації.

Навчання нейронної мережі включає наступні етапи:

Ініціалізація MLContext: MLContext є основним об'єктом, який використовується для роботи з ML.NET. Він забезпечує доступ до функцій створення, тренування та оцінювання моделей машинного навчання.

Підготовка даних: У цьому прикладі вхідні дані представлені в об'єктах TextData, де кожен об'єкт містить текстове поле. Потім ці дані завантажуються у формат IDataView, який є основним форматом для представлення даних у ML.NET.

Створення pipeline: Pipeline - це послідовність операцій, які застосовуються до даних перед тренуванням моделі. У цьому випадку pipeline складається з кількох етапів обробки тексту, таких як токенізація, видалення стоп-слів та використання вбудовування слів.

Тренування моделі: Pipeline тренується на тренувальних даних за допомогою методу Fit, що пристосовує модель до даних.

Трансформація даних: Після тренування модель застосовується до тренувальних та тестових даних за допомогою методу Transform, що перетворює дані у вектори ознак.

Вивід результатів: Результати трансформації перетворюються у масиви для подальшого використання та виводяться для перевірки.

Після цього обрізаємо як тренувальні, так і тестові відгуки. Нижче наведено зразок програмного коду для здійснення цієї операції (див. рисунок 3.2).

```
        // Завантажуємо дані в IDataView
        var trainDataView = mlContext.Data.LoadFromEnumerable(trainData);
        var testDataView = mlContext.Data.LoadFromEnumerable(testData);

        // Визначаємо pipeline для попередньої обробки тексту
        var pipeline = mlContext.Transforms.Text.ProduceWordBags("Features", "Text")
            .Append(mlContext.Transforms.Conversion.MapValueToKey("Label", "Text"))
            .Append(mlContext.Transforms.Text.TokenizeIntoWords("TokenizedText", "Text"))
            .Append(mlContext.Transforms.Text.RemoveDefaultStopWords("CleanedText", "TokenizedText"))
            .Append(mlContext.Transforms.Text.ApplyWordEmbedding("Embeddings", "CleanedText",
                WordEmbeddingEstimator.PretrainedModelKind.GloVeTwitter25D));

        // Тренуємо модель
        var model = pipeline.Fit(trainDataView);

        // Трансформуємо дані
        var transformedTrainData = model.Transform(trainDataView);
        var transformedTestData = model.Transform(testDataView);

        // Перетворюємо в масиви для подальшого використання
        var trainFeatures = mlContext.Data.CreateEnumerable<TransformedData>(transformedTrainData, reuseRowObject: false).ToArray();
        var testFeatures = mlContext.Data.CreateEnumerable<TransformedData>(transformedTestData, reuseRowObject: false).ToArray();

        // Вивід результатів для перевірки
        foreach (var feature in trainFeatures)
        {
            Console.WriteLine(string.Join(",", feature.Features));
        }

        foreach (var feature in testFeatures)
        {
            Console.WriteLine(string.Join(",", feature.Features));
        }
    }
}

2 references
public class TextData
{
    2 references
    public string Text { get; set; }
}

2 references
public class TransformedData
{
    [VectorType(256)]
    2 references
    public float[] Features { get; set; }
}
```

Рисунок 3.2 – Зразок коду обчислення пам’яті

Наступний етап полягає у створенні моделі. Для створення класифікатора всі шари пройшли через процес стеку або накладання шарів:

Перший шар Embedding приймає перекладені в цілі числа слова і знаходить відповідний вектор для кожної пари слово/число. Модель навчається на цих векторах. Розмір векторів збільшує розмір отриманого масиву на 1, що призводить до отримання вимірювання: (партія, послідовність, вкладення);

Наступний шар повертає отриманий вектор заданої довжини для кожного прикладу, усереднюючи розмір ряду. Це дозволяє моделі легко обробляти дані різної довжини;

Цей вектор передається через повнозв'язний шар Dense з 16 прихованими блоками;

Останній шар також є повнозв'язним, але з одним вихідним вузлом. За допомогою функції активації отримується число з плаваючою комою від 0 до 1, що вказує на ймовірність або впевненість моделі.

Нижче наведено приклад програмного коду для побудови моделі нейронної мережі (див. рис. 3.3):

```
    // Завантажуємо дані в IDataView
    var trainDataView = mlContext.Data.LoadFromEnumerable(trainData);
    var testDataView = mlContext.Data.LoadFromEnumerable(testData);

    // Визначаємо pipeline для обробки тексту та створення моделі
    var pipeline = mlContext.Transforms.Text.FeaturizeText("Features", nameof(TextData.Text))
        .Append(mlContext.Transforms.Conversion.MapValueToKey("Label", nameof(TextData.Label)))
        .Append(mlContext.Transforms.Text.TokenizeIntoWords("TokenizedText", "Text"))
        .Append(mlContext.Transforms.Text.RemoveDefaultStopWords("CleanedText", "TokenizedText"))
        .Append(mlContext.Transforms.Text.ApplyWordEmbedding("Embeddings", "CleanedText", WordEmbeddingEstimator.PretrainedModelKind.GloVeTwitter25D))
        .Append(mlContext.Transforms.Concatenate("Features", "Features"))
        .Append(mlContext.Transforms.NormalizeMinMax("Features"))
        .Append(mlContext.BinaryClassification.Trainers.SdcaLogisticRegression());

    // Тренуємо модель
    var model = pipeline.Fit(trainDataView);

    // Оцінюємо модель на тестових даних
    var predictions = model.Transform(testDataView);
    var metrics = mlContext.BinaryClassification.Evaluate(predictions);

    Console.WriteLine($"Accuracy: {metrics.Accuracy:P2}");
}
```

Рисунок 3.3 – Реалізація коду для побудови моделі нейронної мережі

Дана модель має два проміжні шари між вхідним та вихідними даними. Кількість виходів визначає розмір простору представлення у кожному шарі, тобто кількість вільності, яку мережа має під час навчання. Якщо модель має більше прихованих шарів або блоків, то вона може навчитися складнішим концепціям. Однак це може бути витратно з точки зору обчислювальних ресурсів і може

призвести до перенавчання – навчання небажаних патернів, які покращують результати на тренувальних даних, але не на перевірочних. Для навчання моделі необхідно вказати функцію втрат і оптимізатор.

Модель налаштована і продовжується її тренування на тренувальній частині датасету, яка складається з двадцяти п'яти тисяч відгуків, рівною мірою позитивних та негативних.

Тренування моделі розпочинається з 40 епох за допомогою міні-батчів, кожен з яких містить 512 зразків. Це означає, що виконується 40 ітерацій (або проходів) через всі зразки даних у тензорах x_train і y_train. Тут x_train - відгуки на фільми у вигляді числових векторів, а y_train - мітки, які вказують на позитивний чи негативний характер відгуку.

Якщо модель має більше прихованих шарів і/або більше шарів загалом, то нейромережа може вивчити більш складні залежності.

Результати першого десятиліття епох зображено на рисунку 3.4.

```
Epoch 1/40
Epoch 250000/250000 [==============================] - 1s 80us/sample -loss: 0.6876 - acc: 0.6106
Epoch 2/40
Epoch 250000/250000 [==============================] - 1s 80us/sample -loss: 0.6656 - acc: 0.6215
Epoch 3/40
Epoch 250000/250000 [==============================] - 1s 80us/sample -loss: 0.6240 - acc: 0.6616
Epoch 4/40
Epoch 250000/250000 [==============================] - 1s 80us/sample -loss: 0.6011 - acc: 0.6822
Epoch 5/40
Epoch 250000/250000 [==============================] - 1s 80us/sample -loss: 0.5886 - acc: 0.6940
Epoch 6/40
Epoch 250000/250000 [==============================] - 1s 80us/sample -loss: 0.5493 - acc: 0.7096
Epoch 7/40
Epoch 250000/250000 [==============================] - 1s 80us/sample -loss: 0.5216 - acc: 0.7471
Epoch 8/40
Epoch 250000/250000 [==============================] - 1s 80us/sample -loss: 0.4632 - acc: 0.7844
Epoch 9/40
Epoch 250000/250000 [==============================] - 1s 80us/sample -loss: 0.3902 - acc: 0.8257
```

Рисунок 3.4 – Результати тренування нейронної мережі

Після завершення тренування ми перевіряємо ефективність нашої моделі, використовуючи набір даних для перевірки, що складається з 25,000 зразків. Для цього використовується функція evaluate, яка отримує на вхід два масиви: тестові рецензії та відповідні маркування позитивної/негативної рецензії. Результатом цієї операції є пара чисел: відсоток втрат (loss), який вказує на рівень неточності

прогнозів нейронної мережі (чим менше, тим краще), та точність (accuracy), яка оцінює правильність класифікації (див. рис. 3.5).

```
25000/25000 [===
[0.3356132378492, 0.8680]
-
ls 55us/sample loss: 0.3356 - acc: 0.8680
```

Рисунок 3.5 – Результати роботи нейронної мережі на тестових даних

Для кращого уявлення результатів було створено графік втрат та точності на обох етапах – навчання та перевірка моделі, використовуючи вбудовані можливості використаних бібліотек. На горизонтальній осі відображені епохи, на вертикальній – точність для кожної епохи. Точки позначають точність під час навчання, а лінії – точність під час перевірки моделі (див. рис. 3.6).

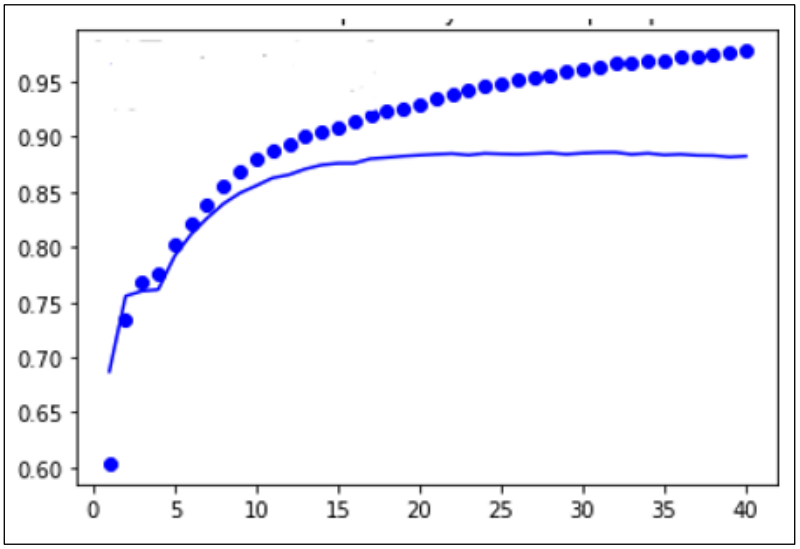


Рисунок 3.6 – Зміна точності залежно від епохи

Під час навчання втрати зменшуються, а точність збільшується з кожною наступною епохою. Проте, після двадцятої епохи точність перестає зростати, що свідчить про перенавчання моделі – коли вона показує кращі результати на даних для навчання, аніж на нових даних для перевірки. Тому має сенс обмежити кількість епох до 20. Результати для 20 епох майже не відрізняються від

результатів для 40 епох. Точність зросла трохи, але це незначна різниця (див. рис. 3.7).

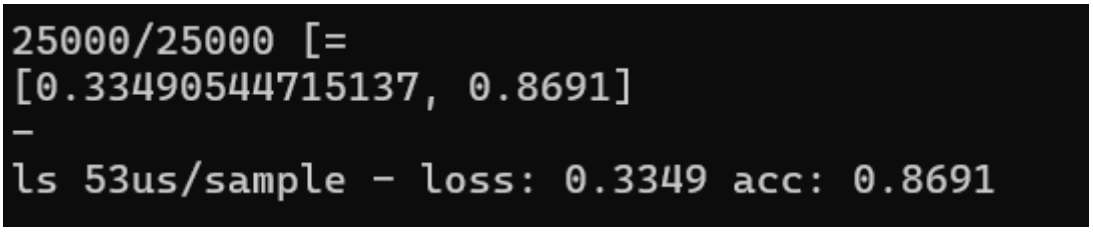


Рисунок 3.7 – Результати для 20 епох

Отже, застосування згорткової нейронної мережі для оцінки настрою у відгуках до фільмів продемонструвало достатньо задовільний результат з точністю на рівні 87%.

3.3 Оцінка ефективності використання LTSM

Для використання LSTM нейронної мережі необхідно підготувати дані для тренування та тестування. Таким чином, до етапів дослідження додаються попередні етапи підготовки даних. Процес дослідження включає наступні кроки:

- обробка даних: перетворення тексту на нижній регістр і видалення розділових знаків;
- токенізація: створення словника слів за їх частотою вживання, де кожне слово у відгуку замінюється на відповідний номер;
- аналіз довжини відгуків та приведення їх до спільної середньої довжини;
- визначення мережевої архітектури LSTM;
- побудова моделі класифікації;
- навчання мережі;
- тестування мережі.

Детально розглянемо етапи 3 та 4, оскільки перші два етапи аналогічні попередньому дослідженню. Для визначення середньої довжини відгуків використовується бібліотека pandas. Нижче наведено приклад програмного коду для визначення середньої довжини відгуків та графічного відображення цих даних у вигляді гістограми (див. рис. 3.8).

```
public static class DataFrameColumnExtensions
{
    0 references
    public static double Mean(this PrimitiveDataFrameColumn<int> column)
    {
        return (double)column.Sum() / column.Length;
    }

    1 reference
    public static double StdDev(this PrimitiveDataFrameColumn<int> column)
    {
        var mean = column.Mean();
        var variance = column.Select(x => Math.Pow((double)(x - mean), 2)).Sum() / (double)column.Length;
        return Math.Sqrt(variance);
    }

    2 references
    public static double Percentile(this PrimitiveDataFrameColumn<int> column, double percentile)
    {
        var sorted = column.OrderBy(x => x).ToArray();
        int N = sorted.Length;
        double n = (N - 1) * percentile + 1;

        if (n == 1d) return (double)sorted[0];
        else if (n == N) return (double)sorted[N - 1];
        else
        {
            int k = (int)n;
            double d = n - k;
            return (double)(sorted[k - 1] + d * (sorted[k] - sorted[k - 1]));
        }
    }
```

Рисунок 3.8 – Реалізація коду для виявлення середньої довжини відгуків

Отже, згідно з отриманими результатами, середня довжина відгуків становить 174 символ. Подібно до попереднього дослідження, відгуки зводяться до однакової загальної довжини. Це перетворення застосовується як до тренувального, так і до тестового набору даних. Тепер ми маємо два набори відгуків однакової довжини, у яких слова кодуються цифрами. Таким чином, дані готові для подальшого використання в нейронних мережах.

Наступним кроком є побудова власної нейронної мережі та визначення класу моделі. Починаючи з визначення гіперпараметрів:

– lstm_size: Кількість одиниць у прихованих шарах LSTM-клітин. Зазвичай вище значення вважається ефективнішим. Типові значення - 128, 256, 512 тощо, але можна використовувати середню довжину відгуку;
– lstm_layers: Кількість шарів LSTM у мережі. Рекомендується почати з одного, а потім додавати при необхідності;
– batch_size: Кількість відгуків, які передаються мережі за один навчальний прохід. Зазвичай рекомендується встановлювати його як можна вище, не перевантажуючи пам'ять;

– learning_rate: Швидкість навчання.

Загальна структура нейронної мережі включає наступні етапи:

– Embedding шар, який перетворює слова (представлені цілими числами) у вектори певного розміру;
– LSTM шар, параметри якого визначаються розмірами схованого стану і кількістю шарів;
– повністю підключений шар, який зводить вихід від LSTM шару до бажаного розміру виводу;
– шар активації сигмоїдів, що перетворює всі вихідні значення до діапазону від 0 до 1;
– вихідні дані: останній вихідний сигмоподібний вектор розглядається як кінцевий вихід мережі.

Після визначення класу моделі (повний код наведено у додатках), створюється екземпляр нейронної мережі і проводиться її навчання на тренувальних даних. Після завершення навчання проводиться перевірка на тестових даних (див. рис. 3.9).

```
var testAcc = new List<float>();

// Load the graph and create a session
var graph = new TFGraph();
var model = File.ReadAllBytes("checkpoints/sentiment_manish.ckpt");
graph.Import(model);
var session = new TFSession(graph);

// Initializing the state
var cellZeroState = graph.OperationByName("cell/zeros_state");
var initialState = session.Run(
    new[] { graph["batch_size"][0], graph["dtype"][0] },
    new[] { TFOutput.OutputList(graph.OperationByName("initial_state/Initialize")) },
    cellZeroState.Outputs
);

// Assuming get_batches is a method that yields batches of test_x and test_y
foreach (var (x, y) in GetBatches(test_x, test_y, batch_size))
{
    var feedDict = new Dictionary<TFOutput, TFTensor>
    {
        { graph["inputs"][0], x },
        { graph["labels"][0], y.reshape(-1, 1) },
        { graph["keep_prob"][0], 1.0f },
        { graph["initial_state"][0], initialState }
    };

    var results = session.Run(
        feedDict.Keys.ToArray(),
        feedDict.Values.ToArray(),
        new[] { graph["accuracy"][0], graph["final_state"][0] }
    );

    var batchAcc = results[0].GetValue<float>();
    initialState = results[1];

    testAcc.Add(batchAcc);
}

Console.WriteLine($"Test accuracy: {testAcc.Average():0.###}");
```

Рисунок 3.9 – Перевірка на тестових даних

Після проведення кількох експериментів виявлено, що нейронна мережа навчається занадто довго, і після близько 20 епох вона починає перенавантажуватися, що призводить до зниження точності передбачень. У зв'язку з цим було вирішено завершити навчання на 15 епохах. Результати тестування представлені на рисунку 3.10.

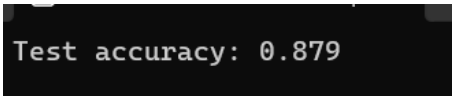


Рисунок 3.10 - Результати тестування для LSTM-мережі

Як видно з результатів, використання LSTM-мережі призвело до незначно кращого результату порівняно з простою згортковою мережею.

3.4 Аналіз використання передової моделі, навченої заздалегідь

Для виявлення тональності текстів ми використовуватимемо попередньо навчену модель Google’s BERT. Оскільки модель вже навчена, нам не потрібно проводити тренування нейронної мережі, і можемо одразу використовувати її для аналізу текстів на настрій. Щоб скористатися цією моделлю, потрібно просто завантажити її з офіційного гітхаб-репозиторія. Модель очікує набір даних у вигляді двох колонок: одна містить відгуки, а інша - їхню полярність. Також треба вказати вхідні параметри (див. рис. 3.11).

```
var mlContext = new MLContext();

// Load data
var dataView = mlContext.Data.LoadFromTextFile<TrainingModel>("C:\\Users\\3622405\\Pictures\\bot\\test.csv", separatorChar: ',');

// Define the pipeline
var pipeline = mlContext.Transforms.Conversion.MapValueToKey("Label")
    .Append(mlContext.Transforms.Text.FeaturizeText("Features", nameof(TrainingModel.Text)))
    .Append(mlContext.MulticlassClassification.Trainers.SdcaMaximumEntropy("Label", "Features"))
    .Append(mlContext.Transforms.Conversion.MapKeyToValue("PredictedLabel"));
```

Рисунок 3.11 - Вхідні параметри

Як і раніше, необхідно провести токенізацію даних: кожне слово у відгуці перетворити на числове значення, а потім представити відгуки у вигляді векторів, так само, як це було зроблено раніше. Для цієї операції використовувалися ті ж методи, що й у дослідженні використання LSTM нейронної мережі.

Лейбли, що вказують на полярність тексту (позитивна або негативна), знову мають бути представлені у вигляді бінарного флагу 0 або 1.

Для отримання результату достатньо лише викликати функцію checkAuc(), передаючи їй набір даних та встановлені параметри. Результати визначення полярності тексту показані на рисунку 3.12.

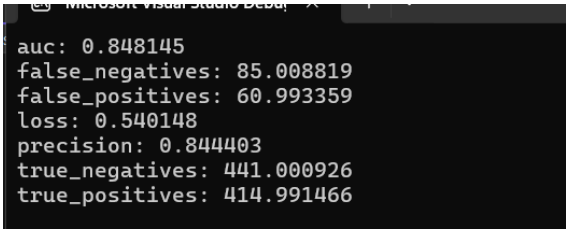


Рисунок 3.12 – Використання BERT для аналізу полярності відгуків

З результатів видно, що точність визначення полярності трохи менша, ніж у попередніх методів. Це пояснюється використанням моделі, яка була попередньо навчена на іншому датасеті. Нейронна мережа не виявила ознак, що специфічні для відгуків на фільми.

3.5 Додаток для визначення емоційного відтінку тексту

Для аналізу настрою тексту, який вводить користувач, використовувались три попередньо навчені нейронні мережі: згорткова, LSTM та модель BERT від Google. Додаток реалізовано у вигляді веб-сторінки з полем для введення тексту та випадаючим списком для вибору моделі. Якщо не вибрано конкретну модель, результат буде показано для всіх трьох варіантів (див. рис. 3.13).

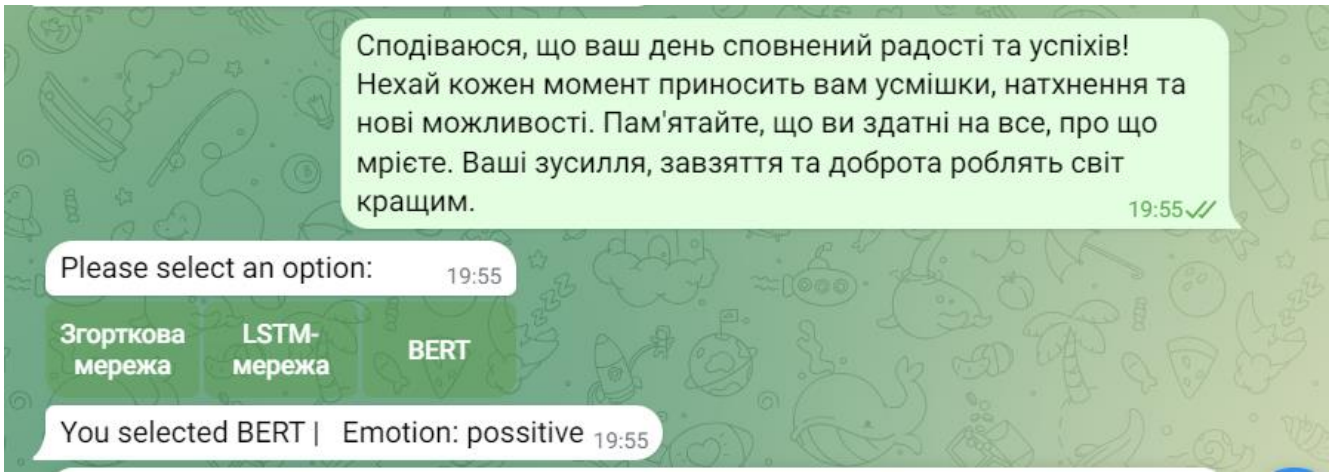


Рисунок 3.13 – Веб-інтерфейс для виявлення полярності тексту

Інтерфейс користувача спроектовано з використанням бібліотеки Telegram API на платформі .NET, та її вбудованих компонент. Для дослідження було використано не лише відгуки на фільми з бази IMDB, а також випадкові тексти з інших датасетів:

- Twitter US Airline Sentiment – цей набір даних містить інформацію з Twitter про авіакомпанії США, зібрану починаючи з лютого 2015 року. Твіти були класифіковані авторами як позитивні, негативні та нейтральні (для цієї роботи використовувалися тільки позитивні та негативні);
- Paper Reviews – у цьому наборі даних зібрані висловлення з наукових публікацій, які відображають позитивні відгуки з міжнародної конференції з обчислювальної техніки та інформатики;
- Amazon Reviews for Sentiment Analysis – цей набір даних містить кілька мільйонів відгуків клієнтів Amazon (вхідний текст) та їх зіркові рейтинги (вихідні мітки) для навчання моделі швидкого аналізу настроїв. Оскільки додаток працює з бінарною класифікацією, відгуки з рейтингом 4-5 вважаються позитивними, а з рейтингом 1-3 – негативними. Нейтральні відгуки не враховуються в цьому наборі даних.

Результати досліджень наведено у таблиці 3.1.

Таблиця 3.1 – Результати досліджень для різних датасетів

Датасет	Згорткова мережа	LSTM-мережа	BERT
Відгуки IMDB	0,86	0,87	0,85
Twitter US Airline Sentiment	0,67	0,701	0,83
Paper Reviews	0,76	0,79	0,84
Amazon Reviews	0,78	0,81	0,84

Як видно з результатів дослідження, власні моделі краще працювали на датасеті, на якому вони були натреновані. Проте, для текстів іншої тематики їх ефективність була нижчою, що можна пояснити тим, що ці моделі тренувалися на меншому корпусі текстів з однієї предметної області. Найгірші результати вони показали з даними з Твіттера, оскільки максимальна довжина повідомлення в Твіттері складає 280 знаків, тоді як середня довжина відгуку на IMDB становить 240 слів.

Найкращі результати з новими датасетами показала нейронна мережа BERT, що пояснюється її тренуванням на менш специфічному датасеті та більшому корпусі текстів.

Однак, розширивши тренувальний датасет для LSTM-мережі, можна потенційно досягти таких самих або навіть кращих результатів. Рекомендується використовувати корпус даних різної довжини та з різних предметних областей.

ВИСНОВКИ

Аналіз настроїв спрямований на автоматичне виявлення в текстах емоційно забарвленої лексики та емоційної оцінки авторів щодо об'єктів, про які йдеться у тексті. Це контекстне витягування тексту, що ідентифікує та витягує суб'єктивну інформацію з вихідного матеріалу.

Аналіз тональності текстів може допомогти зрозуміти закони природної мови і навчити комп'ютер сприймати її на рівні, наближеному до людського. Він також здатний значно покращити якість перекладів.

Цей аналіз допомагає бізнесу зрозуміти соціальні настрої щодо свого бренду, продукту або послуги, контролюючи онлайн-розмови. Його можна використовувати для проведення ринкових досліджень, аналітики продуктів, підтримки користувачів, аналізу зворотного зв’язку та моніторингу соціальних медіа.

Нейронні мережі є потужним інструментом для аналізу тональності текстів. Для аналізу текстів конкретної предметної області (наприклад, звернень до служби підтримки) ефективніше використовувати нейронну мережу, навчану на даних цієї ж області. Якщо ж потрібно аналізувати тональність довільних текстів, для яких не можна виділити спільне джерело або тематику, доцільно використовувати підхід Transfer Learning, застосовуючи попередньо навчені моделі. Дослідження показали, що чим різноманітніші дані для навчання нейронної мережі, тим ефективнішою вона є для аналізу довільних текстів.

У практичних завданнях можуть застосовуватися як теоретичні, так і практичні результати дослідження, оскільки часто виникають завдання як визначення тональності текстів спільного напряму, так і цілком довільних. Отримані натреновані моделі можуть одразу використовуватися для розв’язання завдань аналізу тональності тексту на практиці.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Text & Sentiment Analysis: Key Differences & Real-World Examples URL - https://qualaroo.com/blog/text-analysis-vs-sentiment-analysis-understanding-thedifference/#:~:text=Sentiment%20analysis%20lets%20you%20understand,%2C%20neutral%2C%20sad%2C%20etc (дата звернення: 12.03.2024).

2. Using emotional evaluation of text in a foreign language learning app URL - https://ceur-ws.org/Vol-2899/paper027.pdf (дата звернення: 28.11.2023).

3. The Method of Text Tonality Classification URL - https://www.researchgate.net/publication/348685876_The_Method_of_Text_Tonality_Classification (дата звернення: 28.11.2023).

4. Accuracy vs. precision vs. recall in machine learning: what's the difference? URL - https://www.evidentlyai.com/classification-metrics/accuracy-precision-recall (дата звернення: 28.11.2023).

5. Emotion and Sentiment Analysis: A Practitioner’s Guide to NLP URL - https://www.kdnuggets.com/2018/08/emotion-sentiment-analysis-practitioners-guide-nlp-5.html (дата звернення: 29.11.2023).

6. Best Emotion Detection APIs in 2023 URL - https://www.edenai.co/post/best-emotion-detection-apis (дата звернення: 29.11.2023).

7. Агекян І.А., Рассоха О.В., Грамм О.В. Дослідження методів для розробки програмної системи розпізнавання емоцій та визначення стану здоров'я людини. Біоніка інтелекту. – Харків : ХНУРЕ, – 2020. – № 1 (94). – С. 65-70

8. Artem Khovrat, Volodymyr Kobziev. Using Neural Networks to Identify Fake News. Інформаційні технології та комп’ютерне моделювання"; матеріали статей Міжнародної науково-практичної конференції, м. Івано-Франківськ, 6-10 липня 2023 року. – Івано-Франківськ: п. Голіней О.М., 2023. – с. 79-81

9. Bianchi, F.M., Maiorino, E., Recurrent Neural Networks for Short-Term Load Forecasting, SpringerBriefs – 2017, 72 c.

10. D.S. Nazarenko, Iryna Afanasieva, Nataliia Golian. Investigation of the deep learning approaches to classify emotions in texts. Proceedings of the 5th

International Conference on Computational Linguistics and Intelligent Systems (COLINS-2021), Kharkiv, Ukraine, April 23-24, 2021. – P. 206-225.

11. Simeon Kostadinov. Recurrent Neural Networks with Python Quick Start Guide: Sequential learning and language modeling with TensorFlow. Paperback – 2018, 122 c.

12. Агекян І.А., Рассоха О.В., Грамм О.В. Дослідження методів для розробки програмної системи розпізнавання емоцій та визначення стану здоров'я людини. Біоніка інтелекту. – Харків : ХНУРЕ, – 2020. – № 1 (94). – С. 65-70

13. James McCaffrey, Deep Learning with C# and .NET: Neural Networks and Deep Learning in .NET, Kindle Edition – 2018, 300 c.

14. Ian Goodfellow, Yoshua Bengio, Deep Learning (Adaptive Computation and Machine Learning series), The MIT Press – 2016, 775 c.

15. BERT Explained: State of the art language model for NLP / Towards Data science, URL: https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp- f8b21a9b6270 (дата звернення: 07.04.2024).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ ЗА НАУКОВИМИ НАПРЯМАМИ КЕРІВНИКА ТА НАУКОВЦІВ КАФЕДРИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

7. Агекян І.А., Рассоха О.В., Грамм О.В. Дослідження методів для розробки програмної системи розпізнавання емоцій та визначення стану здоров'я людини. Біоніка інтелекту. – Харків : ХНУРЕ, – 2020. – № 1 (94). –С. 65-70

8. Artem Khovrat, Volodymyr Kobziev. Using Neural Networks to Identify Fake News. Інформаційні технології та комп’ютерне моделювання"; матеріали статей Міжнародної науково-практичної конференції, м. Івано-Франківськ, 6-10 липня 2023 року. – Івано-Франківськ: п. Голіней О.М., 2023. – с. 79-81.

10. D.S. Nazarenko, Iryna Afanasieva, Nataliia Golian. Investigation of the deep learning approaches to classify emotions in texts. Proceedings of the 5th International Conference on Computational Linguistics and Intelligent Systems (COLINS-2021), Kharkiv, Ukraine, April 23-24, 2021. – P. 206-225

12. Агекян І.А., Рассоха О.В., Грамм О.В. Дослідження методів для розробки програмної системи розпізнавання емоцій та визначення стану здоров'я людини. Біоніка інтелекту. – Харків : ХНУРЕ, – 2020. – № 1 (94). – С. 65-70