МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій
Кафедра інтелектуальних технологій

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

НА ТЕМУ

Програмний модуль сентиментального аналізу україномовних відгуків на віртуальному торговельному майданчику

Галузь знань **12 «Інформаційні технології»**
Спеціальність **122 «Комп'ютерні науки»**
Освітня програма **«Комп'ютерні науки»**
Освітній рівень: бакалавр

Виконав: студент 4 курсу, групи КН- 42

Євдокимов Нікіта Віталійович

(прізвище та ініціали)

Керівник: Тмєнова Наталія Пилипівна

(прізвище та ініціали)

кандидат фізико-математичних наук, доцент

(науковий ступінь, звання)

Випускна кваліфікаційна робота бакалавра допущена до захисту
рішенням кафедри *інтелектуальних технологій*
Протокол № 13 від 05.06.2023 р.
зав. кафедри_____ доц. Іларіонов О.Є.

Київ – 2023

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет інформаційних технологій

Кафедра інтелектуальних технологій

Спеціальність 122 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ
Зав. кафедри інтелектуальних технологій

к.т.н., доц. Іларіонов О.Є.
(звання, прізвище та ініціали)

(підпис)

« ___ » _____ 2023 р.

ЗАВДАННЯ
НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Євдокимова Нікіти Віталійовича
(прізвище, ім'я, по батькові)

1. Тема роботи: " Програмний модуль сентиментального аналізу україномовних відгуків на віртуальному торговельному майданчику " затверджена наказом ректора від "___" 2023 року № _____

2.Термін виконання проекту (роботи):): з _____ до _____

3. Вихідні дані до роботи: розробити систему оцінки тональності тексту коментарів на сайті

4. Зміст пояснювальної записки (перелік питань, що підлягають розробці):
1) аналіз предметної області;

2) Проблематика задач визначення тональності тексту та аналіз розроблених датасетів і методів;

3) розробка системи визначення тональності текстів українською мовою.

5. Перелік презентаційного матеріалу (з точним зазначенням обов'язкових презентацій):
мета, об'єкт та предмет (1 слайд), задачі для виконання кваліфікаційної роботи (1 слайд), Приклади використання тональності тексту (1 слайд), Дерево функцій програмного модуля сентиментального аналізу тональності відгуків (3 слайди), діаграма «ЯК БУДЕ» для програмного модуля сентиментального аналізу тональності відгуків (1 слайд), Архітектура комп'ютерної системи обробки текстової інформації (1 слайд), Налаштування розробленого проєкту в Anaconda Navigator (1 слайд), Результат навчання нейронної мережі (1 слайд), Інтерфейс розробленої системи (2 слайди), Тестування розробленої системи визначення тональності тексту відгуків (2 слайди), висновки (1 слайди).

6. Консультанти з випускної кваліфікаційної роботи із зазначенням її розділів, що їх стосуються

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Тмєнова Н.П.		
2	Тмєнова Н.П.		
3	Тмєнова Н.П.		

7. Дата видачі завдання 15 лютого 2023 року

Керівник випускної кваліфікаційної роботи Н.П. Тмєнова

Завдання прийняв до виконання Н.В. Євдокимов

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів	Примітка
1	Обговорення постановки завдання та змісту пояснювальної записки	25.01.2023 -22.02.2023	
2	Вибір та формування теми	23.02.2023 - 26.02.2023	
3	Аналіз предметної області	27.02.2023 – 10.03.2023	
4	Вибір методів рішення задачі	11.03.2023 –20.04.2023	
5	Створення програмного модулю	21.04.2023 -10.05.2023	
6	Оформлення пояснювальної записки	11.05.2023 – 29.05.2023	

Керівник випускної кваліфікаційної роботи Н.П. Тмєнова

Студент Н.В. Євдокимов

Анотація

Євдокимов Нікіта Віталійович виконав випускну кваліфікаційну роботу на тему “Програмний модуль сентиментального аналізу україномовних відгуків на віртуальному торговельному майданчику ” за спеціальністю 122 – «Комп’ютерні науки».

У випускній кваліфікаційній роботі проведено розробку програмного модуля для аналізу тональності тексту в українській мові, яке є важливим інструментом для розуміння емоційного відгуку читачів на різні тексти, включаючи коментарі в інтернеті, відгуки на продукти та послуги, новини тощо. Дослідження також має важливе значення для розвитку сучасних інформаційних технологій, зокрема, в галузі штучного інтелекту та обробки природної мови.

Об'єктом дослідження є аналіз тональності тексту в українській мові, а предметом є методи та алгоритми, що використовуються для визначення позитивної, негативної та нейтральної тональності текстів.

Метою дослідження є вивчення теоретичних аспектів аналізу тональності тексту в українській мові, розробка алгоритмів та методів аналізу, порівняння різних методів аналізу тональності, а також практичне застосування результатів дослідження в галузі обробки природної мови та інформаційних технологій.

Ключові слова: аналіз тональності, обробка природної мови, інформаційні технології, україномовний контент, аналіз відгуків, тональність контенту.

Summary

Yevdokimov Nikita Vitaliyovych has completed his graduation thesis on the topic " Software module for sentimental analysis of Ukrainian-language reviews on a virtual trading platform " in the field of 122 – "Computer Science".

The graduation thesis involved the development of a software module for analyzing the sentiment of text in the Ukrainian language, which is an important tool for understanding the emotional response of readers to different texts, including comments on the internet, product and service reviews, news, and so on. The research is also of significant importance for the development of modern information technologies, particularly in the fields of artificial intelligence and natural language processing.

The object of the research is the analysis of the sentiment of text in the Ukrainian language, while the subject is the methods and algorithms used to determine the positive, negative, and neutral sentiment of texts.

The aim of the research is to study the theoretical aspects of sentiment analysis of text in the Ukrainian language, develop algorithms and methods of analysis, compare different methods of sentiment analysis, and practically apply the research results in the field of natural language processing and information technology.

Keywords: sentiment analysis, natural language processing, information technology, Ukrainian-language content, analysis of reviews, content sentiment.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ 7

ВСТУП 8

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 12

1.1 Аналіз методів визначення тональності текстів 12

1.2 Типи аналізу тональностей 15

1.3 Метрики аналізу тональностей текстів 22

1.4 Аналіз існуючих сервісів оцінки тональності тексту 23

1.5 Постановка задачі 24

Висновки до розділу 1 26

РОЗДІЛ 2 ПРОБЛЕМАТИКА ЗАДАЧ ВИЗНАЧЕННЯ ТОНАЛЬНОСТІ ТЕКСТУ ТА АНАЛІЗ РОЗРОБЛЕНИХ ДАТАСЕТІВ І МЕТОДІВ 27

2.1 Проблематика задач, які вирішуються сервісами оцінки тональності тексту 27

2.2 Аналіз обраних для дослідження методів 39

2.3 Функціональний аналіз програмного модуля сентиментального аналізу тональності відгуків 43

Висновки до розділу 2 47

РОЗДІЛ 3 РОЗРОБКА СИСТЕМИ ВИЗНАЧЕННЯ ТОНАЛЬНОСТІ ТЕКСТІВ УКРАЇНСЬКОЮ МОВОЮ 49

3.1. Архітектура системи та розгортання необхідних компонентів розробки 49

3.2. Аналіз роботи розробленої системи аналізу тональності тексту 58

3.3. Описання процесу навчання моделі 66

Висновки до розділу 3 75

ВИСНОВКИ 77

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 79

ДОДАТОК А 82

Перелік умовних позначень і скорочень

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

API (Application Programming Interface) – прикладний програмний інтерфейс;

XML (*Extensible Markup Language*) – розширена мова розмітки. Текстовий формат для обміну даними;

SQL (*Structured Query Language*) – мова структурованих запитів. Для взаємодії з базами даних;

UI (*User interface*) – графічний інтерфейс користувача;

MVC (*Model-View-Contoller*) – паттерн веб програмування. Складається з моделі сутності, візуального представлення та контроллеру виконання логіки;

SPA (*Single-page application*) – односторінковий додаток. Паттерн веб-програмування;

PHP (*Hypertext Preprocessor*) – мова веб-програмування; *JS* (*Javascript*) – скриптова мова програмування;

Vue (*Vue.js*) – фреймворк мови програмування *Javascript*;

Redis – утіліта зберігання даних у оперативній пам’яті.

Вступ

Актуальність теми. Сучасний рівень розвитку інформаційних технологій зумовлює високу конкуренцію на ринку IT продуктів і послуг, а також змушує фірми-розробники автоматизовувати практично усі бізнес-процеси. Наявність великої кількості даних та широке застосування методів інтелектуального аналізу текстової та графічної інформації, наявність апаратних і програмних засобів зберігання та опрацювання «великих даних» дає змогу майже в автоматичному режимі забезпечувати процеси комунікації між замовниками та розробниками, визначати переваги і недоліки наданих послуг чи проданих товарів, порівнювати конкурентоздатність продуктів, підвищити супровід комп'ютерних систем та їх вдосконалення.

Ця робота важлива для розвитку електронної комерції та покращення якості обслуговування споживачів. Аналіз тональності відгуків може допомогти виробникам та продавцям зрозуміти переваги та недоліки їхніх продуктів, а також виявити та вирішити проблеми, що стосуються якості, обслуговування або інших аспектів.

Крім того, це дослідження сприяє розвитку методів обробки природної мови та аналізу тексту в українській мові. Враховуючи унікальні особливості української мови, розробка системи визначення тональності відгуків є актуальною завданням для забезпечення ефективності та точності результатів аналізу.

Серед найбільш цікавих і популярних методів визначення тональності тексту в українській мові є:

1. Метод машинного навчання з використанням нейронних мереж. Використання нейронних мереж дозволяє побудувати модель, яка автоматично визначає тональність тексту на основі великої кількості навчальних прикладів. Такі моделі можуть враховувати контекст, семантику та інші важливі ознаки тексту для точного визначення його тональності.

2. Метод аналізу словникової тональності. Цей метод використовує словники, де кожному слову присвоєна відповідна емоційна оцінка (позитивна, негативна або

нейтральна). Аналізуючи сукупність слів у тексті, можна визначити загальну тональність шляхом підрахунку сумарної емоційної оцінки.

3. Метод використання машинного перекладу. Цей підхід полягає в перекладі україномовного тексту на англійську або іншу мову, а потім використанні методів аналізу тональності, які розроблені для цільової мови. Після цього, результати можуть бути переведені назад до української мови.

4. Метод використання статистичних моделей. Цей підхід базується на статистичних методах та алгоритмах, які використовуються для аналізу тексту. Він використовує різні статистичні метрики, щоб визначити ймовірність позитивної, негативної або нейтральної тональності тексту.

Ці методи представляють лише деякі з можливих підходів до визначення тональності тексту в українській мові. Комбінація цих методів або розробка нових підходів може привести до ще більш точного та ефективного аналізу тональності українських текстів.

Програмний модуль сентиментального аналізу тональності україномовних відгуків на віртуальному торговельному майданчику надає деякі переваги для електронної комерції:

1. Отримання інформації про сприйняття товарів або послуг: модуль дозволяє аналізувати тон та настрій україномовних відгуків користувачів про товари або послуги на віртуальному торговельному майданчику. Це дозволяє компаніям отримувати інформацію про те, як їхні продукти або послуги сприймаються споживачами.

2. Виявлення проблем та покращення якості: модуль допомагає виявляти негативні відгуки та проблеми, з якими зіштовхуються користувачі. Це дозволяє компаніям оперативно реагувати на незадоволення споживачів та вносити необхідні зміни для покращення якості своїх товарів або послуг.

3. Покращення обслуговування клієнтів: за допомогою модуля компанії можуть аналізувати позитивні відгуки та розуміти, які аспекти їхнього бізнесу особливо сподобалися клієнтам. Це дозволяє вдосконалювати обслуговування та надавати більш індивідуальний підхід до потреб клієнтів.

4. Розробка маркетингових стратегій: модуль допомагає збирати важливі дані про настрій та ставлення споживачів до конкретних продуктів або послуг. Це дозволяє компаніям адаптувати свої маркетингові стратегії, виходячи з потреб та вподобань своїх клієнтів.

Загалом, програмний модуль сентиментального аналізу тональності україномовних відгуків надає електронній комерції цінну інформацію про сприйняття продуктів або послуг споживачами, допомагає виявляти проблеми та покращувати якість, підвищує задоволення клієнтів та сприяє розвитку ефективних маркетингових стратегій.

Дослідженню методів аналізу тональності і класифікації текстової інформації присвячено ряд наукових робіт як українських, так і закордонних вчених, зокрема Дж. Мак-Кіна, Г. Болла, Д. Холла, Г. Ланса, У. Уільямсона, Н. Джардайна, Є.М. Бравермана, А.А. Дорофеюка, І.Б. Мучника, М.Г. Загоруйко та ін. Сьогодні ефективно застосовуються засоби аналізу тональності тексту, розроблені фірмами *Google* та *Yahoo*.

На сьогодні розроблено багато методів для вирішення задач щодо визначення тональності тексту. Однак, ефективність їх застосування залежить від природи вхідних даних, області та способів використання. Обґрунтування і розробка методу визначення тональності текстової інформації при вдосконаленні комп'ютерних систем є однією з нових та актуальних задач і вимагає додаткового дослідження існуючих методів і засобів.

Метою роботи є дослідження методів і засобів визначення тональності текстової інформації для вдосконалення комп'ютерних систем.

Для досягнення вказаної мети в роботі поставлено наступні задачі:

– аналіз наукових праць і практик реалізації методів і засобів виявлення та аналізу тональності текстової інформації;

– аналіз особливостей опрацювання відгуків користувачів при вдосконаленні комп'ютерних систем;

– обґрунтування інтелектуальних методів аналізу та визначення тональності у відгуках користувачів про комп'ютерні системи;

– розробка інтелектуального методу виявлення тональності відгуків користувачів;

– розробка програмного засобу виявлення та оцінювання тональності текстової інформації при супроводі комп’ютерних систем.

Об’єкт дослідження: процес виявлення тональності текстової інформації.

Предмет дослідження: методи і засоби виявлення та аналізу тональності текстової інформації.

Методи дослідження. Для вирішення поставлених задач використано наступні методи: аналіз та узагальнення – при проведенні аналізу існуючих методів і засобів виявлення тональності текстової інформації; теорії імовірності та математичної статистики, машинного навчання – при розробці методу виявлення та аналізу тональності відгуків користувачів комп’ютерних систем; проектування та програмування – при розробці програмного засобу збору, аналізу та виявлення тональності текстової інформації при супроводі комп’ютерних систем; експеримент і тестування – для апробації розробленого методу та програмного засобу.

Розділ 1
Аналіз предметної області

1.1 Аналіз методів визначення тональності текстів

Аналіз тональності тексту (сентимент-аналіз, англ. *Sentiment analysis,* англ. *Opinion mining*) – це процес виявлення та визначення емоційної валентності текстового матеріалу, зокрема, визначення позитивної, негативної або нейтральної емоційної оцінки, яку він виражає [1]. Цей процес включає аналіз тексту, його змісту, семантики, синтаксису та контексту з метою розуміння його емоційної сутності.

Аналіз тональності тексту може застосовуватися в різних галузях, включаючи соціальні медіа, маркетинг, відгуки споживачів, політичний аналіз та інші сфери, де важливо розуміти, як сприймається інформація аудиторією. Цей процес може здійснюватися як за допомогою ручного аналізу людьми, так і з використанням автоматизованих методів, таких як машинне навчання та обробка природної мови, для автоматичного визначення та класифікації тональності текстових даних.

Аналіз сентиментів може бути змодельований як проблема класифікації, де метою є визначення тональності тексту та його призначення до певного класу – позитивного, негативного або нейтрального.

У рамках класифікації, модель сентимент-аналізу отримує на вхід текст і повинна прийняти рішення про його тональність. Вона може використовувати різні підходи, такі як машинне навчання або статистичні методи, для розподілу текстів до відповідних класів на основі їхньої емоційної оцінки.

У процесі моделювання проблеми класифікації аналізу сентиментів, можуть використовуватися різноманітні підходи, такі як навчання з учителем або навчання без учителя. При навчанні з учителем, модель використовує набір навчальних даних, де кожен текст має відому мітку класу (позитивний,

негативний або нейтральний). Вона вивчає закономірності та шаблони в цих даних для подальшого класифікації нових текстів.

Крім того, можуть використовуватися методи векторизації тексту, де текст перетворюється на числовий вектор, що відображає його особливості та зміст.

Це дозволяє моделі аналізувати та розрізняти різні аспекти тексту, такі як слова, фрази або контекстуальні залежності, для здійснення більш точного визначення тональності.

Отже, моделювання проблеми класифікації дозволяє побудувати ефективні моделі аналізу сентиментів, які можуть автоматично визначати тональність текстів і використовуватися для різних задач, таких як моніторинг соціальних мереж, відгуків користувачів або аналіз публічної думки.

Типи думок можуть впливати на тональність тексту, оскільки різні види думок можуть мати різну емоційну оцінку. Нижче наведено кілька типів думок та їх можливий вплив на тональність:

1. Позитивні думки: Ці думки виражають задоволення, радість, захоплення або позитивне ставлення до чого-небудь. Вони можуть підвищувати позитивну тональність тексту і сприяти виникненню позитивних асоціацій.
2. Негативні думки: Ці думки виражають незадоволення, роздратування, обурення або негативне ставлення до чого-небудь. Вони можуть знижувати загальну тональність тексту і створювати негативні асоціації.
3. Нейтральні думки: Ці думки не виражають сильну емоційну оцінку і можуть бути безпосередньо пов'язані з фактами, об'єктивними спостереженнями або висловлюваннями без явного емоційного відтінку. Вони можуть мати незначний вплив на загальну тональність тексту.

Важливо враховувати, що визначення тональності тексту є складним завданням і залежить від контексту, розуміння семантики та використаних методів аналізу. Однак, розпізнавання типів думок може допомогти розуміти, як емоційна оцінка впливає на загальну тональність тексту.

Визначення тональності відгуку в залежності від типу думок може залежати від підходів і методів, які використовуються для аналізу тексту. Декілька способів визначення тональності відгуку враховуючи типи думок [3]:

1. Підхід на основі словника: Словниковий підхід включає створення словників або лексичних баз, де кожному слову призначається емоційна оцінка (позитивна, негативна або нейтральна). При аналізі відгуку, слова в тексті порівнюються зі словником, і їхня емоційна оцінка враховується для визначення загальної тональності. Залежно від типу думок (позитивних, негативних або нейтральних), відповідні емоційні оцінки будуть впливати на загальну тональність відгуку.
2. Машинне навчання з учителем: В цьому підході, модель навчається на позначених даних, де кожен відгук має відповідну мітку класу (позитивний, негативний або нейтральний). Модель вивчає закономірності і зв'язки між текстовими ознаками і їхніми емоційними оцінками, що дозволяє їй передбачати тональність нових відгуків на основі їхньої семантики та контексту. У цьому випадку, типи думок будуть представлені як класи, до яких відносяться відповідні емоційні оцінки.
3. Використання правил та шаблонів: Інший підхід полягає в розробці правил або шаблонів, які визначають типи думок і їх вплив на тональність. Наприклад, можуть бути встановлені правила, що описують, що певні типи думок (наприклад, позитивні фрази, негативні заяви) впливають на загальну тональність відгуку. Застосування таких правил та шаблонів дозволяє враховувати типи думок при визначенні тональності тексту.

Ці підходи можуть бути використані окремо або комбіновано для визначення тональності відгуку, враховуючи типи думок. Важливо враховувати, що визначення типів думок та їх впливу на тональність може бути залежним від домену, мови та контексту, тому розробка відповідних правил або навчання моделі на відповідних даних є важливими кроками для досягнення точного аналізу тональності.

1.2 Типи аналізу тональностей

Сучасні програми для аналізу тональності тексту використовують різні типи аналізу для отримання більш точних та комплексних результатів. Деякі з них включають:

1. Аналіз на рівні слова: в цьому типі аналізу програми оцінюють емоційну валентність окремих слів або фраз у тексті. Кожному слову або фразі присвоюється певна емоційна оцінка (позитивна, негативна або нейтральна). В результаті отримується сумарна емоційна оцінка на основі аналізу всіх слів у тексті.
2. Аналіз на рівні речення: в цьому типі аналізу програми враховують структуру речень та взаємозв'язки між словами у тексті. Вони виявляють синтаксичні залежності та аналізують емоційну валентність речень, щоб визначити загальну тональність.
3. Аналіз на рівні контексту: в цьому типі аналізу програми враховують контекст, в якому використовується текст. Вони розуміють семантичні зв'язки, залежності та асоціації між словами та фразами, що допомагає зрозуміти справжнє значення тексту та визначити його тональність.
4. Машинне навчання: багато програм для аналізу тональності використовують методи машинного навчання для побудови моделей, які навчаються на великому обсязі навчальних даних. Ці моделі визначають емоційну валентність тексту на основі вивчених залежностей та шаблонів. Вони можуть враховувати різні ознаки тексту, включаючи слова, синтаксичні структури, контекстуальні залежності тощо.
5. Глибинний аналіз: деякі програми використовують глибинний аналіз для аналізу тональності тексту. Це означає, що програми старанно вивчають текст на рівні смислу та семантики, а не просто враховують окремі слова або фрази. Вони використовують методи обробки природної мови (Natural Language Processing – NLP), щоб розуміти контекст, виявляти абстрактні поняття, розбирати складні конструкції та ідентифікувати семантичні залежності в тексті.

Глибинний аналіз може включати використання методів, таких як синтаксичний аналіз, семантичний аналіз, аналіз залежностей та інші для отримання більш деталізованого розуміння тексту і визначення його емоційної валентності. Цей підхід дозволяє програмам розрізняти ситуації, деталізувати емоційні відтінки, враховувати імпліцитні смисли та здійснювати більш точну класифікацію тональності тексту.

Глибинний аналіз є більш складним та обчислювально витратним, але він може забезпечити більш докладну та розширену інформацію про емоційну валентність тексту, допомагаючи отримати більш глибоке розуміння реакції аудиторії та контекстуального значення текстової інформації.

У сучасних системах автоматичного визначення емоційної оцінки тексту досить поширеним підходом є використання одномірного емотивного простору [1]. Цей підхід базується на класифікації тексту за однією основною емоційною оцінкою, яка може бути позитивною або негативною.

У такому одномірному емотивному просторі тексти класифікуються на дві категорії: позитивні та негативні. Позитивна оцінка відображає позитивні емоції, такі як радість, задоволення, захоплення, тоді як негативна оцінка відображає негативні емоції, такі як незадоволення, обурення, сумнів або гнів.

Одномірний емотивний простір використовується через свою простоту та легкість застосування. Він дозволяє швидко класифікувати тексти за загальним тоном, що дозволяє отримати загальне уявлення про емоційну валентність тексту.

Однак, варто зазначити, що такий підхід не враховує всі відтінки та складнощі емоційної сфери. Він спрощує аналіз, але може бути обмежений в здатності визначати більш складні емоційні стани та субтільні відтінки в тексті.

Найпоширеніші алгоритми для визначення емоційної оцінки тексту в одномірному емотивному просторі використовують методи машинного навчання, лексичні бази або комбінацію обох. Вони навчаються на позначених даних, де кожен текст має мітку позитивної або негативної емоційної оцінки, і застосовують ці знання для класифікації нових текстів.

Можна класифікувати полярність документа по багатосмуговій шкалою, що було зроблено Пангом і Снайдером [2]. Панг і Снайдер розробили методологію класифікації полярності документа за допомогою багатосмугової шкали. Цей підхід дозволяє визначати не просто позитивну або негативну тональність документа, але й оцінювати його емоційну валентність на кілька рівнів або відтінків.

Замість традиційного бінарного підходу до класифікації (позитивний або негативний), багатосмугова шкала дозволяє присвоїти документу значення від -5 до +5, де -5 відповідає найбільш негативній полярності, +5 – найбільш позитивній полярності, а 0 – нейтральній полярності.

Цей підхід дозволяє деталізувати класифікацію і враховувати більш широкий спектр емоцій та відтінків, які можуть бути присутні у тексті. Наприклад, документ може мати значення +2, що вказує на помірну позитивну полярність, або -3, що вказує на помірну негативну полярність.

Цей підхід знаходить застосування в різних областях, де точність і деталізація аналізу емоційної валентності є важливими, наприклад, в аналізі соціальних мереж, маркетингових дослідженнях, аналізі відгуків користувачів та інших сферах, де важливо отримати більш точне розуміння емоційного тону тексту.

Інший дослідницький напрямок у сфері аналізу тексту – це ідентифікація суб'єктивності та об'єктивності текстів [4]. Цей напрямок спрямований на визначення ступеня суб'єктивності або об'єктивності висловлювань, що містяться у тексті.

Суб'єктивність вказує на наявність емоційних, особистих, суб'єктивних оцінок або поглядів у тексті. Такі висловлювання можуть відображати особисті переконання, враження, думки або суб'єктивні оцінки автора. Наприклад, фраза "Цей фільм був дуже захоплюючим" є суб'єктивною, оскільки вона виражає особисту оцінку автора про фільм.

З іншого боку, об'єктивність вказує на відсутність особистих оцінок, емоцій або суб'єктивних думок у тексті. Такі висловлювання зазвичай базуються на фактах, об'єктивних спостереженнях або наукових даних. Наприклад, фраза "За останні

10 років температура підвищилася на 1,5 градуси Цельсія" є об'єктивною, оскільки вона передає фактичну інформацію без суб'єктивних оцінок.

Ідентифікація суб'єктивності/об'єктивності в тексті може здійснюватися за допомогою різних методів, включаючи машинне навчання, аналіз природної мови та використання лексичних баз або правил. Дані методи дозволяють виявити особливості семантики, лексики та синтаксису тексту, що вказують на наявність суб'єктивних або об'єктивних ознак.

Цей дослідницький напрямок має значення в різних областях, таких як медіааналітика, аналіз соціальних мереж, розпізнавання фейків, судові експертизи та інші, де розуміння суб'єктивності та об'єктивності тексту може мати важливе значення для вивчення публічної думки, фактів або оцінок.

Модель більш докладного аналізу називається аналізом на основі функції/аспекту (aspect-based sentiment analysis) і є підхідом до аналізу тональності тексту, який дозволяє докладніше розглянути відгуки або тексти, визначаючи емоційну оцінку для конкретних аспектів або функцій [5].

У цій моделі текст розглядається як колекція речень або фраз, і для кожного аспекту або функції відгуку окремо визначається емоційна оцінка. Аспекти можуть бути пов'язані з різними аспектами продукту, послуги або сутності, які згадуються у тексті.

Наприклад, при аналізі відгуку про смартфон можуть бути визначені такі аспекти як якість камери, продуктивність, дизайн, екран тощо. Для кожного з цих аспектів відгуку окремо визначається його емоційна оцінка, наприклад, позитивна, негативна або нейтральна.

Модель аналізу на основі функції/аспекту може використовувати методи машинного навчання, правилові системи або комбінацію обох для визначення емоційної оцінки для кожного аспекту. Для цього можуть використовуватися лексичні бази, словники, семантичні правила та інші методи для ідентифікації конкретних аспектів та їхньої емоційної валентності.

Аналіз на основі функції/аспекту дозволяє отримати більш деталізовану інформацію про емоційну оцінку для окремих аспектів або функцій, що

допомагає користувачам отримати більш конкретну інформацію про те, як певні аспекти впливають на загальну емоційну тонованість тексту. Це може бути особливо корисно в ситуаціях, коли важливо розуміти емоційну реакцію аудиторії на конкретні аспекти продукту, послуги або події.

Існує багато видів аналізу настроїв, а інструменти аналізу тональності варіюються від систем, які зосереджуються на полярності (позитивні, негативні, нейтральні), до систем, які виявляють почуття і емоції (сердиті, щасливі, сумні тощо) або визначають наміри (наприклад, зацікавлені в. не зацікавлений).

1. Аналіз полярності: цей тип аналізу фокусується на визначенні полярності тексту, тобто визначенні, чи є відгук позитивним, негативним або нейтральним. Це основна форма аналізу тональності, де текст класифікується за основною емоційною оцінкою.

2. Аналіз емоцій: цей тип аналізу спрямований на виявлення конкретних емоцій або почуттів, виражених у тексті. Він може розпізнавати такі емоції, як щастя, сум, радість, обурення, страх, подяка тощо. Такий аналіз надає більш глибоке розуміння емоційного стану автора тексту.

3. Аналіз намірів: цей тип аналізу спрямований на виявлення намірів, що містяться в тексті. Наприклад, визначення, чи є автор зацікавленим або не зацікавленим у певному питанні, продукті або послузі. Це дозволяє отримати інформацію про ставлення автора до обговорюваної теми.

4. Аналіз соціальних сигналів: цей тип аналізу використовує сигнали зі соціальних медіа, такі як лайки, коментарі, репости, для визначення загальної реакції аудиторії на певний вміст або подію. Він надає інформацію про популярність, прийняття або неприйняття певного вмісту від групи користувачів.

5. Контекстуальний аналіз: цей тип аналізу враховує контекст, в якому використовується текст. Він враховує семантичні зв'язки, ідіоми, культурні нюанси та інші контекстуальні фактори, щоб краще розуміти справжнє значення і тональність тексту.

Кожен з цих підходів має свої переваги і може використовуватися в залежності від конкретних потреб аналізу тональності тексту. Деякі системи можуть комбінувати різні методи для отримання більш комплексної та точної інформації про емоційну валентність тексту.

Дрібнозернистий аналіз (fine-grained analysis) в контексті аналізу тональності тексту означає розподіл класифікації на більш деталізовані категорії або підкатегорії. Він передбачає розширення стандартних класифікаційних категорій (наприклад, позитивний, негативний, нейтральний) на більш конкретні класи, що дозволяє отримати більш докладну інформацію про емоційну валентність тексту.

У дрібнозернистому аналізі можуть використовуватися класи, які відображають відтінки або нюанси емоцій. Наприклад, замість просто "позитивного" класу можуть використовуватися підкатегорії, такі як "дуже позитивний", "трохи позитивний", "легко позитивний" і так далі. Так само, для "негативного" класу можуть бути використані підкатегорії, що відображають різні рівні негативності. Дрібнозернистий аналіз дозволяє виражати більш точні емоційні нюанси та відтінки, які можуть бути присутні у тексті. Це особливо корисно, коли потрібно отримати детальну інформацію про сприйняття аудиторією, оцінки певних аспектів або відгуків. Дрібнозернистий аналіз може використовуватися в різних галузях, таких як соціальні медіа, маркетинг, відгуки користувачів, аналіз настроїв тощо, де важлива деталізація емоційного аналізу.

Деякі системи аналізу тональності також надають різні відтінки полярності, визначаючи, чи позитивні або негативні настрої асоціюються з певним почуттям. Це означає, що системи розрізняють не тільки загальну емоційну оцінку тексту, але й конкретні почуття, які пов'язані з цією оцінкою.

Наприклад, система може визначити, що позитивний настрій в тексті асоціюється з почуттями радості, захоплення, задоволення або подяки. У той же час, негативний настрій може асоціюватися з почуттями обурення, суму, розчарування або гніву. Це дозволяє отримати більш диференційовану

інформацію про емоційну тонованість тексту, розкриваючи більше деталей про те, які саме почуття пов'язані з позитивними або негативними висловлюваннями. Цей підхід дає змогу краще розуміти відтінки емоцій та сприйняття, що може бути важливо в контексті аналізу відгуків, соціальних медіа, реклами та інших джерел інформації. Він дозволяє краще враховувати різноманітність людських емоцій та їхній контекстуальний характер, забезпечуючи більш глибоке та точне розуміння емоційного стану, який виражений у тексті.

Багатомовний аналіз настроїв може бути складним завданням з кількох причин. По-перше, існує велика мовна різноманітність, і кожна мова має свою унікальну лексику, граматику та культурні відтінки. Це вимагає розробки моделей та алгоритмів, які можуть працювати ефективно з різними мовами. По-друге, емоційні відтінки та сприйняття настроїв можуть значно варіюватися в залежності від культурного контексту. Одне й те ж висловлювання може мати різну емоційну валентність в різних культурах або серед різних соціальних груп. Це вимагає урахування культурних особливостей під час багатомовного аналізу настроїв. Крім того, амбігвітність та залежність емоційного вираження від контексту можуть ускладнювати багатомовний аналіз настроїв. Деякі слова або фрази можуть мати кілька можливих емоційних відтінків, і визначення правильного контексту може бути важливим для точного розуміння настроїв.

Багатомовний аналіз настроїв вимагає обробки великого обсягу текстових даних з різних мов, що вимагає ефективних алгоритмів, потужних обчислювальних ресурсів та методів обробки масштабних даних для забезпечення швидкої та точної аналітики.

Незважаючи на ці складнощі, багатомовний аналіз настроїв є важливим напрямком дослідження, оскільки дозволяє краще розуміти емоційні реакції людей в різних культурах та мовах і використовувати цю інформацію у різних сферах, включаючи соціальні медіа, маркетинг, огляди продуктів і послуг та інші аспекти суспільного життя.

1.3 Метрики аналізу тональностей текстів

Метрики аналізу тональності текстів використовуються для оцінки результатів алгоритмів та моделей, що визначають тональність текстових документів. Основні метрики, що використовуються в такому контексті, включають:

1. Точність (Accuracy): Ця метрика вимірює відсоток правильно класифікованих документів від загальної кількості документів. Вона вказує на загальну ефективність системи.
2. Точність позитивного та негативного класів (Precision and Recall): Precision вимірює, який відсоток документів, визначених системою як позитивні (або негативні), є дійсно позитивними (або негативними). Recall вимірює, який відсоток фактично позитивних (або негативних) документів був правильно визначений системою. Ці метрики дозволяють оцінити точність і повноту системи.
3. F-міра (F-measure): Це гармонічне середнє між точністю і повнотою. Вона використовується для більш об'єктивної оцінки систем, особливо коли точність і повнота мають різні значення.
4. ROC-крива (Receiver Operating Characteristic curve) і AUC-ROC (Area Under the ROC Curve): ROC-крива відображає залежність між True Positive Rate (TPR) і False Positive Rate (FPR) для різних порогових значень класифікатора. AUC-ROC вимірює площу під ROC-кривою і використовується для оцінки ефективності класифікатора.
5. Матриця помилок (Confusion Matrix): Матриця помилок відображає кількість правильно і неправильно класифікованих документів для кожного класу. Вона дозволяє детально проаналізувати результати класифікації.

Ці метрики допомагають оцінити ефективність систем аналізу тональності текстів і дозволяють порівнювати різні підходи та моделі на основі їх виконання. Зазвичай використовуються комбінації декількох метрик для більш повного оцінювання системи.

1.4 Аналіз існуючих сервісів оцінки тональності тексту

Для дослідження роботи існуючих сервісів оцінки тональності тексту, було проведено аналіз кількох популярних систем, що використовуються в сучасному контексті. Основними критеріями оцінки були точність, швидкодія, масштабованість та функціональні можливості.

Один з досліджених сервісів – Natural Language Processing API від Google Cloud, відомий своєю високою точністю та розширеними функціональними можливостями. Він забезпечує аналіз тональності тексту, виявлення ключових слів та іменованих сутностей, а також розрізнення мови та переклад тексту.

Ще одним розглянутим сервісом є Microsoft Azure Text Analytics, який також пропонує аналіз тональності тексту, виявлення ключових фраз та мовний переклад. Він володіє високою швидкодією та масштабованістю, а також надає можливість роботи з текстом у багатьох мовах.

Іншим сервісом, що було досліджено, є IBM Watson Natural Language Understanding. Цей сервіс надає можливість аналізувати тональність тексту, виявляти ключові поняття, ентітети та відносини між ними. Він також пропонує розпізнавання мови та переклад тексту.

Усі досліджені сервіси виявилися ефективними і надійними у визначенні тональності тексту. Вони забезпечують широкі можливості для аналізу, а також підтримують роботу з різними мовами. Однак, варто зазначити, що точність і результати можуть варіюватися в залежності від контексту і складності текстів.

Загальною висновкою з дослідження є те, що існують потужні сервіси оцінки тональності тексту, які можуть бути використані для різноманітних завдань, включаючи аналіз соціальних медіа, моніторинг брендів, відгуки користувачів та інші сфери. При виборі сервісу важливо враховувати конкретні потреби та вимоги проекту, а також здійснювати тестування для оцінки його відповідності.

1.5 Постановка задачі

Метою роботи є дослідження існуючих методів аналізу тональності текстів.

Було поставлено наступні задачі для випускної кваліфікаційної роботи:

1. Дослідити сучасний стан методів та технологій аналізу тональності тексту в українській мові.
2. Розглянути різні підходи до аналізу тональності, включаючи правила-базований підхід, методи машинного навчання та глибинного навчання.
3. Зробити огляд і порівняння існуючих систем та сервісів оцінки тональності тексту, які працюють з українською мовою.
4. Вивчити основні проблеми, з якими стикаються сучасні системи оцінки тональності тексту в українській мові, такі як мовна різноманітність, культурні контексти та амбігвітність.
5. Розробити програмний модуль сентиментального аналізу, який здатний визначати тональність україномовних текстів на основі навчаної моделі.
6. Зібрати та підготувати відповідний набір даних для навчання та тестування моделі аналізу тональності.
7. Навчити модель, використовуючи підготовлені дані, для досягнення оптимальної точності та надійності визначення тональності.
8. Провести експериментальне порівняння розробленого програмного модуля з існуючими системами оцінки тональності в українській мові.
9. Оцінити результати експерименту та провести аналіз переваг та обмежень розробленого модуля.
10. Зробити висновки щодо ефективності та можливих напрямків подальшого розвитку аналізу тональності тексту в українській мові.

Ці задачі спрямовані на дослідження сучасних підходів та технологій аналізу тональності тексту, розробку програмного модуля для визначення тональності україномовних текстів, а також на експериментальне порівняння та оцінку

результатів. Це дозволить зрозуміти сильні сторони та обмеження існуючих систем та внести внесок у розвиток цієї області досліджень.

Також в рамках дослідження буде розроблено веб-додаток, який надасть користувачам можливість використовувати сервіс оцінки тональності тексту. Додаток буде мати наступні можливості:

1. Введення тексту: Користувачі зможуть ввести текст, який вони бажають проаналізувати на тональність. Це може бути відгук про товар, коментар до новини або будь-який інший текст.
2. Аналіз тональності: Після введення тексту, додаток буде застосовувати розроблену модель аналізу тональності для визначення позитивної, негативної або нейтральної тональності тексту. Результат аналізу буде відображений користувачеві.
3. Візуалізація результатів: Додаток буде надавати графічну візуалізацію результатів аналізу, наприклад, за допомогою діаграм або кольорових позначок, що допоможе користувачеві зрозуміти тональність тексту на перший погляд.
4. Збереження та відстеження історії: Користувачі зможуть зберігати результати аналізу та відстежувати їх історію. Це дозволить повертатись до попередніх аналізів та порівнювати результати.
5. Можливість налаштування: Додаток може надати користувачам можливість налаштувати параметри аналізу, наприклад, вибрати конкретну модель або встановити власні параметри.
6. Інтеграція з іншими сервісами: Веб-додаток може мати можливість інтеграції з іншими сервісами або платформами, такими як соціальні мережі або електронна комерція, щоб користувачі могли використовувати аналіз тональності для своїх потреб.

Цей веб-додаток буде надавати користувачам зручний і простий спосіб отримати аналіз тональності тексту та зробити об'єктивні висновки на основі результатів.

Висновки до розділу 1

У даному розділі проведено аналіз сучасного стану проблеми оцінки тональності текстів і поставлені задачі для дослідження сервісів оцінки текстової інформації. Досліджено напрями обробки природньої мови та її підвиду – оцінку тональності тексту. Оглянуто ряд задач, які вирішуються застосуванням сучасних сервісів, що базуються на вищезгаданих технологіях. Системи активно використовуються для розв'язання бізнес-задач та забезпечують підвищення ефективності розвитку бізнесу.

Проведено огляд основних критичних моментів, які негативно впливають на рівень точності сучасних систем оцінки тональності тексту. У цих напрямках активно проводяться дослідницькі роботи, але проблеми цілком ще не вирішені.

Проведено дослідження роботи сервісів оцінки тональності тексту, які широко представлені на сьогоднішній день для досягнення бізнес-цілей. Це дозволило оцінити ефективність представлених систем і оцінити їх коректність у реальних умовах.

За результатами проведеного аналізу було поставлено задачі задачі для випускної кваліфікаційної роботи та визначено основний напрямок використання додатку для визначення тональності україномовних відгуків.

РОЗДІЛ 2
ПРОБЛЕМАТИКА ЗАДАЧ ВИЗНАЧЕННЯ ТОНАЛЬНОСТІ ТЕКСТУ ТА АНАЛІЗ РОЗРОБЛЕНИХ ДАТАСЕТІВ І МЕТОДІВ

2.1 Проблематика задач, які вирішуються сервісами оцінки тональності тексту

Проблематика, яку вирішують сервіси оцінки тональності тексту, включає низку важливих задач і викликів, пов'язаних з аналізом та інтерпретацією тональності текстової інформації. Основні проблеми включають наступне:

1. Семантична амбігвітність: текст має складну структуру і може містити багато різних значень та інтерпретацій. Проблема полягає в тому, що одне й те саме слово або фраза може мати різний смисл залежно від контексту. Наприклад, слово "хороший" може мати позитивне значення, але у певному контексті може бути використане з негативною інтонацією.
2. Мовна різноманітність: мовна різноманітність становить виклик для систем оцінки тональності, оскільки різні мови мають свою синтаксичну та семантичну структуру. Врахування особливостей конкретної мови, такої як українська, вимагає розробки специфічних методів і моделей, а також підтримки відповідних ресурсів.
3. Обробка великого обсягу даних: з ростом кількості текстових даних, які генеруються щодня, виникає потреба в ефективних алгоритмах із масштабованістю для швидкої та точної обробки цих даних. Аналіз тональності тексту повинен бути здатний працювати з великим обсягом даних в реальному часі.
4. Врахування контексту: однією з важливих проблем є врахування контексту при визначенні тональності тексту. Значення та інтерпретація тексту можуть змінюватись залежно від контексту, включаючи специфічні ситуації, культурні особливості та індивідуальні переконання.
5. Розрізнення між суб'єктивністю та об'єктивністю: оцінка тональності тексту пов'язана з розрізненням між суб'єктивними висловлюваннями, що

відображають особисту думку або емоції, та об'єктивними фактами. Важливо розробляти алгоритми, які здатні відрізняти ці два типи висловлювань для точнішого визначення тональності.

6. Нерівномірний розподіл тональностей: текстові дані можуть мати нерівномірний розподіл тональностей, де більшість текстів можуть бути нейтральними, а лише невелика частка містить позитивну або негативну тональність. Ефективні методи повинні бути здатні враховувати цей нерівноважний розподіл та надавати точні результати навіть у таких умовах.

Ці проблеми створюють виклики для розробників сервісів оцінки тональності тексту, і їх вирішення вимагає досліджень, розробки нових алгоритмів та моделей, а також збільшення обсягу та якості текстових даних для навчання та валідації.

2.1.1 Можливості використання аналізу тональності тексту

Аналіз тональності тексту відкриває широкі можливості в різних сферах застосування. Ось детальний огляд можливостей використання аналізу тональності тексту:

1. Соціальні медіа моніторинг: аналіз тональності тексту дозволяє відстежувати та аналізувати відгуки, коментарі та думки користувачів в соціальних медіа. Це допомагає компаніям відстежувати свою репутацію, виявляти негативні відгуки та проблеми, а також виявляти можливості для поліпшення продуктів та послуг.

2. Клієнтський сервіс та підтримка: аналіз тональності тексту дозволяє компаніям відстежувати задоволеність клієнтів, виявляти проблеми та пропозиції, а також швидко реагувати на них. Це допомагає покращувати якість обслуговування та забезпечувати задоволення клієнтів.

3. Маркетинг та реклама: Аналіз тональності тексту дозволяє оцінити ефективність рекламних кампаній, виявляти потенційні проблеми зі сприйняттям повідомлень, а також виявляти ключові слова та теми, які найбільше впливають на сприйняття цільової аудиторії.

4. Фінансовий аналіз: аналіз тональності тексту може бути використаний для моніторингу фінансових новин та прогнозування ринкової поведінки. Він допомагає виявити ключові фактори, що впливають на ціни акцій, валютні курси та інші фінансові показники.

5. Публічна думка та політичний аналіз: аналіз тональності тексту може бути використаний для вимірювання публічної думки щодо політичних подій, виборів, законодавчих ініціатив та інших суспільно важливих питань. Це допомагає політикам та аналітикам розуміти настрої громадськості та приймати відповідні рішення.

6. Моніторинг конкурентів: аналіз тональності тексту дозволяє відстежувати та аналізувати відгуки про конкурентів та їх продукти. Це допомагає виявляти сильні та слабкі сторони конкурентів, а також знаходити можливості для покращення власних продуктів та послуг.

Це лише декілька прикладів можливостей використання аналізу тональності тексту. За допомогою цього аналітичного інструменту можна отримати цінну інформацію, яка допомагає у прийнятті рішень, вдосконаленні продуктів та послуг, а також покращенні спілкування з клієнтами та виявленні нових можливостей.

1. Можливості використання аналізу тональності тексту в задачі розвитку бренду

Аналіз тональності тексту має значний потенціал для використання в задачах розвитку бренду. Детальніше розглянемо можливості використання аналізу тональності тексту в цьому контексті:

1. Моніторинг репутації бренду: аналіз тональності тексту дозволяє відстежувати та аналізувати відгуки, коментарі, повідомлення у соціальних медіа та інших джерелах про ваш бренд. Це допомагає виявляти негативні відгуки та проблеми, а також визначати позитивні аспекти та сильні сторони бренду. Такий моніторинг дозволяє зрозуміти сприйняття бренду клієнтами та реагувати на них вчасно.

2. Аналіз конкурентів: аналіз тональності тексту дозволяє відстежувати та аналізувати відгуки, коментарі та повідомлення про конкурентів. Це допомагає

виявити переваги та недоліки конкурентів, а також знайти можливості для покращення власного бренду. Аналізування сприйняття конкурентів допомагає визначити свої унікальні переваги та розвивати стратегії конкурентного позиціонування.

3. Кампанії управління репутацією: аналіз тональності тексту може бути використаний для виявлення позитивних та негативних відгуків про бренд у реальному часі. Це дозволяє реагувати на негативні ситуації та проблеми швидко і ефективно. Крім того, аналіз може виявити ключові питання та скарги клієнтів, що дозволяє прийняти відповідні заходи для вирішення цих питань і покращення репутації бренду.

4. Виявлення тенденцій та потреб клієнтів: аналіз тональності тексту допомагає виявити тенденції, настрої та потреби клієнтів, враховуючи їх відгуки та коментарі. Це дозволяє адаптувати продукти, послуги та маркетингові стратегії під вимоги та очікування клієнтів, покращуючи їх задоволеність і лояльність до бренду.

5. Ефективність рекламних кампаній: аналіз тональності тексту може бути використаний для оцінки ефективності рекламних кампаній. Шляхом аналізу відгуків та реакцій на рекламу, можна зрозуміти, які елементи та повідомлення сприймаються позитивно, а які викликають негативну реакцію. Це допомагає покращувати стратегії реклами та залучати більше клієнтів до бренду.

6. Оцінка задоволеності клієнтів: аналіз тональності тексту може бути використаний для вимірювання та оцінки задоволеності клієнтів. Відгуки та коментарі допомагають зрозуміти, як клієнти сприймають продукти, послуги та взаємодію з брендом. Це дозволяє ідентифікувати проблеми та можливості для покращення задоволеності клієнтів та збільшення їх лояльності до бренду. В роботі [27] було проведено аналіз настроїв щодо епізодів сервіалу «Гри престолів» (рис. 2.1).

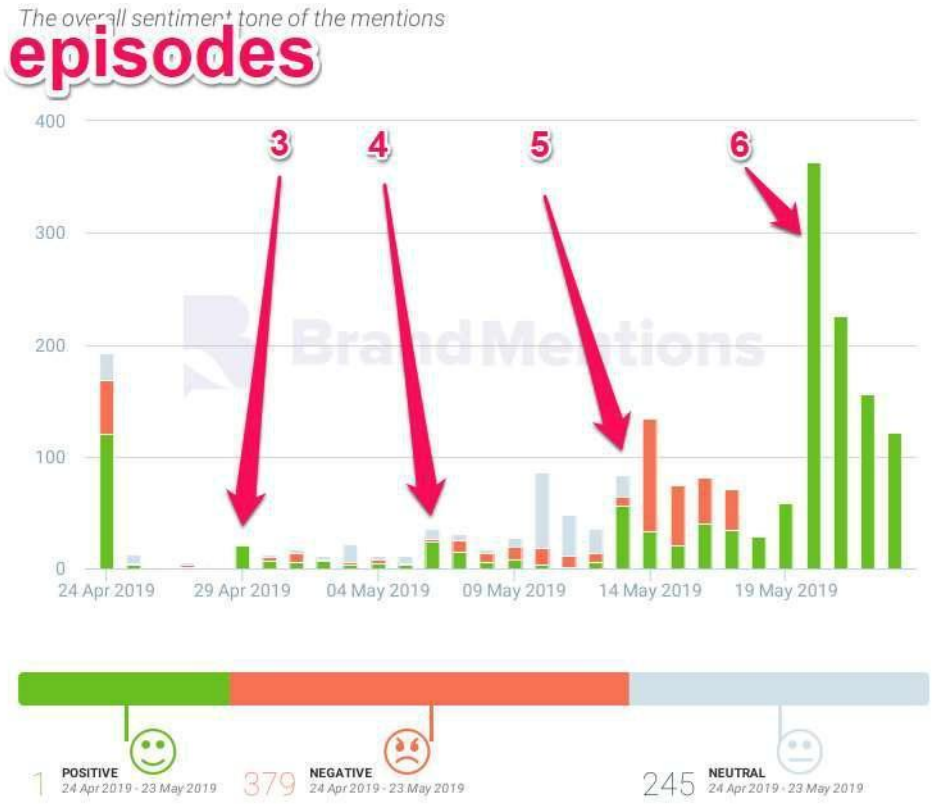


Рисунок 2.1 – Аналіз настроїв за сезонами серіалу [27]

2. Можливості використання аналізу тональності тексту в Задачі дослідження ринку

Аналіз тональності тексту має значний потенціал для використання в задачах дослідження ринку. Детальніше розглянемо можливості використання аналізу тональності тексту в цьому контексті:

1. Виявлення сегментів ринку: аналіз тональності тексту дозволяє виявити різні сегменти ринку та їх настрої. Він допомагає визначити, які групи клієнтів позитивно або негативно реагують на певний продукт або послугу. Це дозволяє компаніям налаштувати свою стратегію маркетингу та звернути увагу на потенційно прибуткові сегменти ринку.

2. Оцінка конкурентної ситуації: аналіз тональності тексту допомагає оцінити сприйняття конкурентів на ринку. Він дозволяє виявити, які конкуренти отримують позитивні або негативні відгуки, як їх продукти або послуги сприймаються ринком. Це дозволяє знайти конкурентні переваги та можливості для покращення власних продуктів або послуг.

3. Виявлення потреб ринку: аналіз тональності тексту допомагає виявити потреби та побажання ринку. Він дозволяє виявити ключові слова, фрази та теми, які зустрічаються найчастіше в позитивних або негативних відгуках. Це допомагає компаніям адаптувати свою продукцію, послуги та маркетингові стратегії під потреби ринку та забезпечити більш ефективне задоволення клієнтів.

4. Визначення маркетингових трендів: аналіз тональності тексту дозволяє виявити маркетингові тренди та популярні теми серед споживачів. Він допомагає визначити, які продукти, послуги або концепції отримують більше позитивних відгуків та підтримки. Це дозволяє компаніям виявляти нові можливості та розвивати інноваційні продукти або послуги, що відповідають вимогам ринку.

5. Моніторинг маркетингових кампаній: аналіз тональності тексту дозволяє оцінювати ефективність маркетингових кампаній та промо-акцій. Він допомагає вимірювати реакцію споживачів на рекламні повідомлення та акції, виявляти негативні аспекти або потенційні проблеми. Це дозволяє покращити стратегію маркетингу, залучити більше клієнтів та збільшити конверсію.

Застосування аналізу тональності тексту в задачах дослідження ринку дозволяє компаніям краще розуміти споживачів, виявляти можливості для покращення продукції та маркетингових стратегій, а також підвищувати задоволеність та лояльність клієнтів. Так в роботі [28] показано, як аналіз настроїв може надати додатковий вимір зусиллям роботи компанії (рис. 2.2).

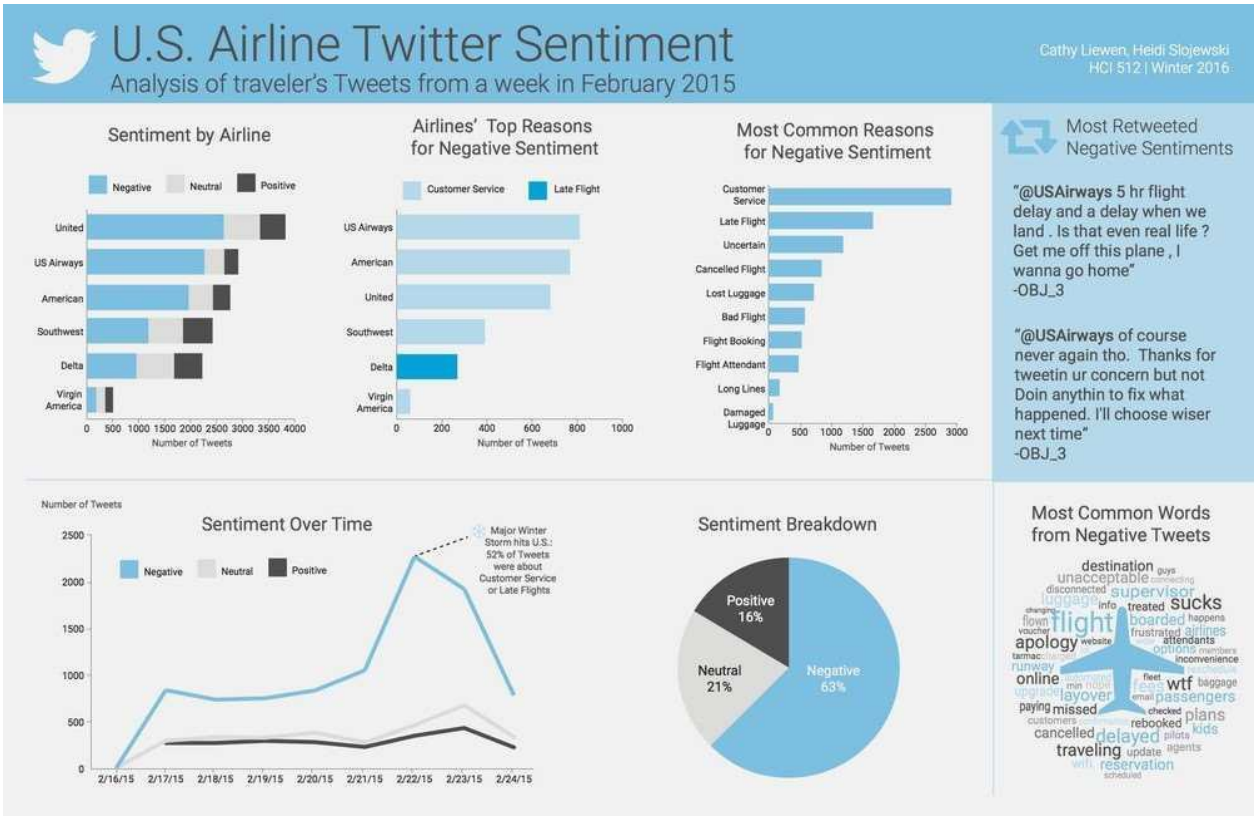


Рисунок 2.2 – Аналіз тональності інформаційних джерел про авіакомпанію [28]

3. Можливості використання аналізу тональності тексту в задачі аналізу конкурентів

Аналіз тональності тексту надає розширені можливості для використання в задачах аналізу конкурентів. Детальніше розглянемо ці можливості:

1. Відстеження реакції на продукти та послуги конкурентів: Аналіз тональності тексту дозволяє відстежувати та аналізувати відгуки, відгуки користувачів та інші текстові матеріали, що стосуються продуктів та послуг конкурентів. Це дозволяє отримати інформацію про сприйняття та задоволеність споживачів від конкурентів, їх сильні та слабкі сторони. Такий аналіз допомагає виявити можливості для покращення власних продуктів та послуг, а також розробити ефективні стратегії конкурентної боротьби.

2. Визначення конкурентного позиціонування: Аналіз тональності тексту дозволяє оцінювати сприйняття та увагу, що приділяється конкурентам у порівнянні з власним брендом. Він допомагає виявити, як конкуренти сприймаються ринком, їхні сильні та слабкі сторони, а також визначити свої

унікальні переваги і вирішити, як ефективно конкурувати. Такий аналіз дозволяє визначити власну позицію на ринку і розробити стратегії конкурентного позиціонування.

3. Моніторинг маркетингових активностей конкурентів: Аналіз тональності тексту дозволяє виявити та оцінити маркетингові активності конкурентів, включаючи рекламні кампанії, акції та промо-заходи. Це допомагає відстежувати реакцію ринку на активності конкурентів та оцінювати їхню ефективність. Такий моніторинг дозволяє компанії адаптувати свої маркетингові стратегії, залучати більше клієнтів та збільшувати конкурентоспроможність.

4. Виявлення нових можливостей: Аналіз тональності тексту дозволяє виявити нові можливості на ринку, пов'язані зі змінами в сприйнятті конкурентів або реакцією ринку на їхні продукти та послуги. Це дозволяє компанії бути в курсі останніх тенденцій та потреб ринку та вчасно реагувати на них, щоб отримати перевагу перед конкурентами.

Аналіз тональності тексту є потужним інструментом для аналізу конкурентного середовища. Використовуючи його можливості, компанії можуть отримати цінні інсайти про своїх конкурентів, розробити ефективні стратегії конкурентного позиціонування та виявити нові можливості для розвитку свого бізнесу [30].

4. Можливості використання аналізу тональності тексту в Задачі аналізу продукту

Аналіз тональності тексту має великий потенціал для використання в задачах аналізу продукту. Детальніше розглянемо можливості використання аналізу тональності тексту в цьому контексті:

1. Оцінка задоволеності клієнтів: аналіз тональності тексту дозволяє оцінити задоволеність клієнтів від певного продукту. Шляхом аналізу відгуків, коментарів та оцінок клієнтів, можна виявити позитивні та негативні аспекти продукту, проблеми та сильні сторони. Це допомагає компаніям розуміти, як їхні продукти сприймаються споживачами і як можна поліпшити їхні характеристики для більшої задоволеності клієнтів.

2. Виявлення та усунення проблем: аналіз тональності тексту допомагає виявити проблеми, які можуть виникати з продуктом. Негативні відгуки та коментарі клієнтів допомагають ідентифікувати недоліки, дефекти або проблеми з функціональністю продукту. Це дозволяє компаніям вжити відповідних заходів для усунення проблем і поліпшення якості продукту.

3. Розробка стратегій вдосконалення продукту: Аналіз тональності тексту допомагає виявити побажання, пропозиції та ідеї клієнтів щодо поліпшення продукту. Позитивні відгуки та коментарі можуть містити цінні пропозиції, які допоможуть удосконалити продукт або додати нові функції, що відповідають потребам клієнтів. Це дозволяє компаніям розробити стратегії вдосконалення продукту та забезпечити його більшу конкурентоспроможність на ринку.

4. Моніторинг конкурентів: аналіз тональності тексту допомагає відстежувати реакцію ринку на продукти конкурентів. Шляхом аналізу відгуків і коментарів клієнтів про продукти конкурентів можна зрозуміти, як вони сприймаються ринком і які їхні сильні та слабкі сторони. Це дозволяє компаніям виявляти переваги свого продукту в порівнянні з конкурентами та розробляти ефективні стратегії конкурентної боротьби.

5. Прогнозування попиту: аналіз тональності тексту допомагає в прогнозуванні попиту на певний продукт. Шляхом аналізу відгуків та коментарів клієнтів можна оцінити, як позитивно або негативно сприймається продукт на ринку, що може свідчити про потенційний попит. Це дозволяє компаніям планувати виробництво та маркетингові стратегії залежно від попиту та забезпечувати більшу ефективність виробництва та збуту.

Використання аналізу тональності тексту в задачах аналізу продукту допомагає компаніям отримувати цінні інсайти щодо задоволеності клієнтів, виявляти проблеми та можливості для вдосконалення продукту, моніторити конкурентів та прогнозувати попит на ринку. Це дозволяє компаніям бути більш конкурентоспроможними, задовольняти потреби клієнтів та покращувати якість своїх продуктів [31].

5. Можливості використання аналізу тональності тексту в Задачі аналізу настрою персоналу

Аналіз тональності тексту має значний потенціал для використання в задачах аналізу настрою персоналу. Розглянемо детальніше можливості використання аналізу тональності тексту в цьому контексті:

1. Виявлення загального настрою: аналіз тональності тексту дозволяє виявити загальний настрій співробітників на основі їхніх письмових відгуків, коментарів або повідомлень. Це дозволяє оцінити загальну задоволеність та ставлення персоналу до роботи і організації. За допомогою аналізу тональності тексту можна виявити позитивні, негативні та нейтральні аспекти настрою персоналу.

2. Виявлення проблем та конфліктів: аналіз тональності тексту допомагає виявити проблеми, конфлікти або негативні аспекти в настрої персоналу. Негативні відгуки, коментарі або повідомлення можуть вказувати на проблеми в робочому середовищі, конфлікти з колегами або незадоволення з певними аспектами роботи. Виявлення цих проблем дозволяє реагувати на них вчасно і приймати відповідні заходи для вирішення та поліпшення робочого клімату.

3. Оцінка емоційного стану: аналіз тональності тексту може допомогти в оцінці емоційного стану персоналу. Відгуки, коментарі та повідомлення можуть відображати різні емоції, такі як радість, розчарування, стрес, задоволення тощо. Розуміння емоційного стану персоналу може бути корисним для компаній, дозволяючи їм адаптувати політику кадрів, створювати сприятливі умови для праці та забезпечувати психологічний комфорт у робочому оточенні.

4. Виявлення позитивних аспектів та мотивації: аналіз тональності тексту дозволяє виявити позитивні аспекти і мотивацію персоналу. Позитивні відгуки, коментарі або повідомлення можуть свідчити про задоволеність роботою, сприятливі умови або високий рівень мотивації. Виявлення цих аспектів може допомогти компаніям визначити, що сприяє успіху та задоволеності персоналу, і планувати подальші заходи для їх збереження та розвитку.

Використання аналізу тональності тексту в задачах аналізу настрою персоналу дозволяє компаніям отримати глибокі інсайти щодо настрою, задоволеності та

емоційного стану персоналу. Це допомагає покращувати робоче середовище, забезпечувати задоволеність та мотивацію співробітників, а також вчасно виявляти проблеми і конфлікти, сприяючи покращенню робочої ефективності та результативності.

2.1.2 Визначення основних проблемних задач сучасних сервісів оцінки тональності тестів

Визначення проблематики сучасних сервісів оцінки тональності текстів включає наступні аспекти:

1. Проблема виявлення сарказму: Сарказм є складним елементом комунікації, де семантичне значення висловлювання протилежне його літеральному значенню. Для багатьох сервісів оцінки тональності тексту виявлення сарказму є викликом. Це пов'язано з необхідністю розпізнавання інтонації, контексту та вживання іронічних слів або висловів. Недоцільне виявлення сарказму може призводити до неточностей в оцінці тональності тексту.

Чотири типи сарказму Елізабет Кемп [30]: пропозиційний ("Це виглядає як ідеальний план!"), Вбудований ("я люблю, коли мене ігнорують"), "префікс" ("Як ті хлопці вірять слову, яке вони говорять") і невербальні "(знизує плечима) Насправді дуже корисно!" (рис. 2.3).

2. Проблема виявлення заперечень: Заперечення в тексті можуть змінювати його семантичне значення та тон. Розпізнавання заперечень є важливим в контексті аналізу тональності, оскільки вони можуть викликати протилежну реакцію у споживачів. Виявлення заперечень вимагає розуміння контексту та логіки речення. Неправильне розпізнавання заперечень може призводити до неточностей в визначенні тональності тексту.

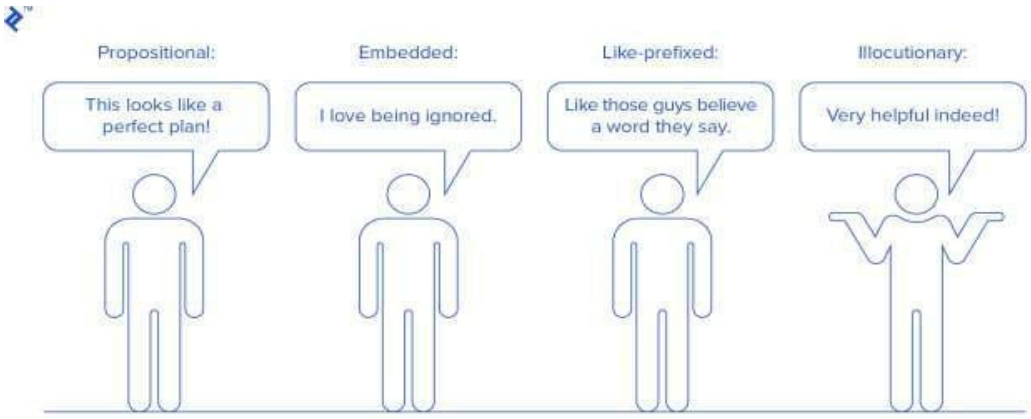


Рисунок 2.3 – Приклади сарказму

3. Проблема неоднозначності слів: Багатозначні слова є ще однією складністю при аналізі тональності тексту. Одне й те ж слово може мати різні значення в залежності від контексту. Наприклад, слово "гарячий" може мати позитивне значення у зв'язку з їжею, але негативне значення у зв'язку з погодою. Розуміння контексту та використання лексичних баз або семантичних моделей можуть допомогти вирішити цю проблему.

4. Проблема багатополярності: Багатополярність відображає наявність кількох тональних аспектів у тексті. Текст може мати одночасно позитивні, негативні та нейтральні відгуки або оцінки. Деякі сервіси оцінки тональності можуть бути обмежені в розпізнаванні таких багатополярних виразів і просто призначати загальну тональність тексту. Для більш точної оцінки необхідно розвивати алгоритми, які враховують багатополярність тексту і визначають значення для кожного аспекту окремо.

Визначення цих проблем допомагає усвідомити складнощі, з якими стикаються сервіси оцінки тональності тексту та відповідні розділи досліджуваної роботи будуть присвячені розробці методів та алгоритмів, які зможуть вирішити ці проблеми та покращити точність оцінки тональності тексту.

2.2 Аналіз обраних для дослідження методів

2.2.1 Згорткові нейронні мережі

Згорткова нейронна мережа (*convolutional neural network, CNN*) – це класс нейронних мереж глубокого навчання, що зазвичай використовується для аналізу зображень, відео, а також обробки природних мов [12]. Згорткова нейронна мережа є регуляризованною версією багатошарового персептрона, який розроблено таким чином, щоб для роботи нейронної мережі потрібно було проводити мінімальну попередню обробку.

Робота згорткової нейронної мережі зазвичай є переходом від конкретних особливестей вхідних даних до більш абстрактних деталей, і далі до ще більш абстрактних деталей, доходячи до виділення понять високого рівня. Мережа є самоналаштованною і самостійно виробляє необхідну ієрархію абстрактних ознак чи послідовностей ознак, проводячи фільтрацію неважливих деталей і виділяючи важливі.

Ознаки, які виробляє нейронна мережа, зазчивай є доволі складними для розуміння, тому у випадку, якщо система ігнорує якісь істотні ознаки, замість зміни змісту ознак рекомендується удосконалити структуру та архітектуру мережі.

У звичайному перцептроні, який представляє собою повнозв'язну нейронну мережу, кожен нейрон пов'язаний з усіма нейронами попереднього шару, причому кожна зв'язок має свій персональний ваговий коефіцієнт [12].

У згортковій нейронній мережі в операції згортки використовується лише обмежена матриця ваг невеликого розміру, яку «рухаються» по всьому оброблюваного шару (на самому початку – безпосередньо по вхідним даним), де формують після кожного зсуву сигнал активації для нейрона наступного шару з аналогічною позицією (рис. 2.4).

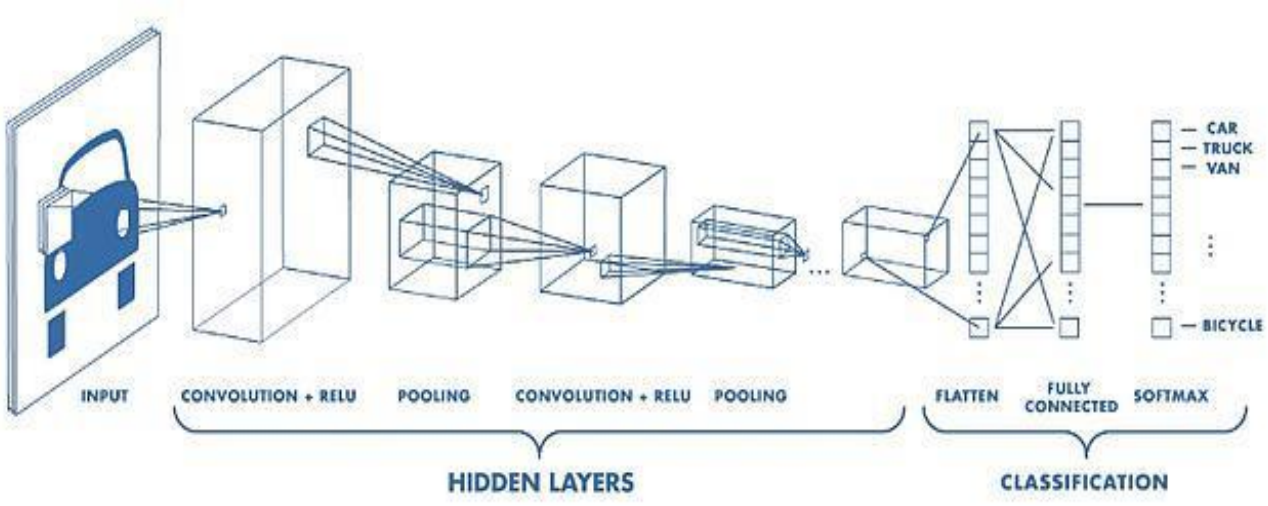


Рисунок 2.4 – Типова архітектура згорткової нейронної мережі

Тобто для різних нейронів вихідного шару використовуються одна і та ж сама матриця ваг, яку також часто називають ядром згортки. Її інтерпретують як графічне кодування якої-небудь ознаки, наприклад, наявність похилої лінії під певним кутом чи наявність певних фігур, що наприклад повторюються на різних зображеннях. Тоді наступний шар, що вийшов в результаті операції згортки такою матрицею ваг, показує наявність даної ознаки в оброблюваному шарі і її координати, формуючи так звану карту ознак (англ. *Feature map*). При цьому такі ядра згортки не закладаються дослідником заздалегідь, а завжди формуються самостійно шляхом навчання мережі класичним методом зворотного поширення помилки.

Природно, в згортковій нейронній мережі набір ваг не один, а ціла гама, що кодує елементи вхідних даних (наприклад лінії і дуги під різними кутами для зображення).

При цьому такі ядра згортки не закладаються дослідником заздалегідь, а формуються самостійно шляхом навчання мережі класичним методом зворотного поширення помилки.

Прохід кожним набором ваг формує свій власний примірник карти ознак, роблячи нейронну мережу багатоканальної (багато незалежних карт ознак на

одному шарі). Також слід зазначити, що при переборі шару матрицею ваг її пересувають зазвичай не на повний крок (розмір цієї матриці), а на невелику відстань. Так, наприклад, при розмірності матриці ваг 5 × 5 її зрушують на один або два нейрона (пікселя) замість п'яти, щоб не «переступити» шукану ознаку.

Метод навчання з учителем (на маркованих даних) є одним з найбільш простих і популярних способів навчання моделей аналізу тональності тексту. Цей метод базується на використанні маркованих даних, де для кожного текстового прикладу відомо його очікувана тональність.

Один з основних методів навчання з учителем, що застосовується для аналізу тональності тексту, це метод зворотного поширення помилки (backpropagation) і його модифікації. Цей метод використовується для навчання нейронних мереж, які є популярними моделями для аналізу тональності.

Процес навчання з використанням методу зворотного поширення помилки включає такі кроки:

1. Ініціалізація вагів: початкові ваги нейронної мережі встановлюються випадковим чином або за допомогою попереднього навчання.
2. Пряме поширення: вхідні дані (текстові приклади) передаються через нейронну мережу, де вони проходять через шари нейронів і відповідні ваги.
3. Обчислення помилки: для кожного прикладу обчислюється помилка між очікуваною та фактичною виходами мережі.
4. Зворотнє поширення помилки: Помилка поширюється назад через мережу, і ваги оновлюються, щоб мінімізувати помилку.
5. Оновлення ваг: за допомогою градієнтного спуску ваги нейронів оновлюються, залежно від розміру помилки.
6. Повторення: кроки 2-5 повторюються для всього навчального набору даних до тих пір, поки помилка не буде достатньо мала або поки не буде досягнуто заданої точності.

Цей метод може бути модифікований, наприклад, застосовуючи різні активаційні функції, регуляризацію для уникнення перенавчання або

використання глибоких нейронних мереж для здатності до складних аналізування тексту.

Метод навчання зворотного поширення помилки і його модифікації є ефективними і широко використовуваними способами навчання моделей аналізу тональності. Вони дозволяють моделі вчитися на основі маркованих даних та вирішувати завдання класифікації тональності з високою точністю.

2.2.2 Рекурентна нейронна мережа – довга та короткочасна пам’ять

Рекурентна нейронна мережа (*Recurrent neural network*; *RNN*) – клас нейронних мереж, в якому міжвузлові з’єднання утворюють направлену у часі послідовність (орієнтований граф) [14]. Завдяки цьому рекурентні нейронні мережі надають можливість опрацьовувати послідовні просторові ланцюги чи серії подій у часі. Рекурентні мережі, на відміну від багатошарових перцептронів, можуть використовувати свою внутрішню пам’ять для обробки послідовностей довільної довжини. Через це подібні нейронні мережі є широко застосовними в задачах розпізнавання тексту чи мови, а також для задач обробки природних мов [15], де вони дозволяють широко використовувати попередній досвід.

Найбільш розповсюдженими варіантами архітектури рекурентної нейронної мережі (RNN) для аналізу тональності тексту є:

1. Проста RNN (Simple RNN): це базовий тип RNN, в якому кожен нейрон має зв'язки зі своїми попередніми стани. Вона може бути використана для аналізу послідовностей тексту, де кожен елемент в послідовності впливає на наступні.
2. LSTM (Long Short-Term Memory): LSTM є вдосконаленим варіантом RNN, який вирішує проблему зникнення градієнту. Вона здатна зберігати та використовувати довгострокову залежність між елементами послідовності, що робить її ефективною для аналізу тональності тексту.
3. GRU (Gated Recurrent Unit): GRU є ще одним варіантом RNN, який розв'язує проблему зникнення градієнту. Вона має менше параметрів, ніж LSTM, що

робить її швидшою у навчанні та використанні. GRU також здатна вирішувати проблему збереження довгострокової залежності.

4. Bidirectional RNN: це тип RNN, в якому інформація передається в обох напрямках – вперед і назад по послідовності тексту. Це дозволяє моделі враховувати контекст з обох сторін і поліпшує точність аналізу тональності.

Ці варіанти архітектури рекурентних нейронних мереж є широко використовуваними в аналізі тональності тексту, оскільки вони здатні враховувати залежності між словами в послідовності та вирішувати завдання класифікації тональності з високою точністю.

2.3 Функціональний аналіз програмного модуля сентиментального аналізу тональності відгуків

Огляд систем сентиментального аналізу тональності відгуків дозволив побудувати дерево функцій для програмного модуля сентиментального аналізу тональності україномовних відгуків на віртуальному торговельному майданчику (рис. 2.5):

1. Збір та передача даних:

– Визначення джерел даних (наприклад, відгуки на платформі електронної комерції).

– Збір відгуків та пов'язаної інформації, такої як рейтинги, дати, автори тощо.

– Передача даних для подальшого аналізу.

2. Обробка даних:

2.1. Попередня обробка даних:

– Токенізація: розбиття тексту на окремі токени (слова, фрази).

– Прибирання зайвої інформації, такої як пунктуація, числа, спеціальні символи.

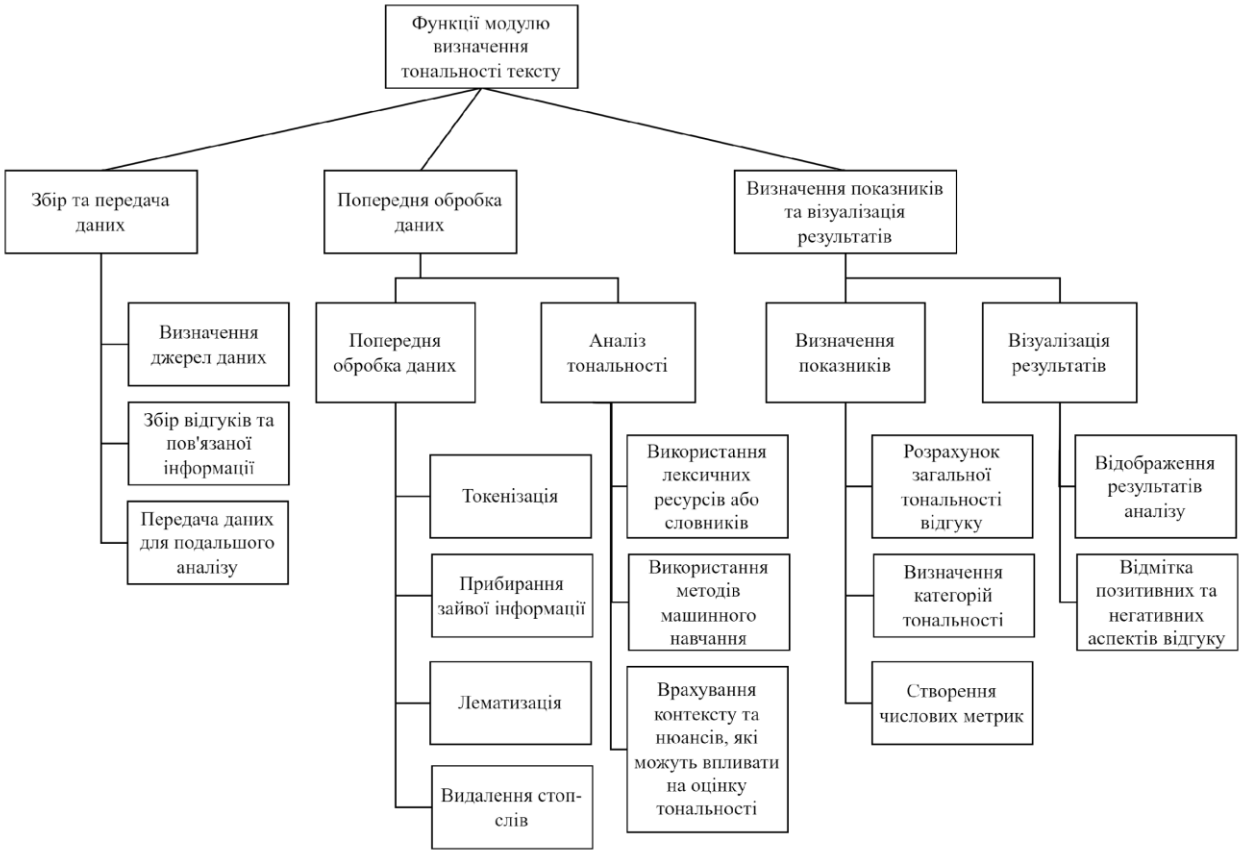


Рисунок 2.5 – Дерево функцій програмного модуля сентиментального аналізу тональності відгуків

– Лематизація або стемінг: зведення слів до їх базової форми.

– Видалення стоп-слів: видалення загальних та непродуктивних слів, які не несуть значущої інформації (наприклад, "і", "у", "що").

2.2. Аналіз тональності:

– Використання лексичних ресурсів або словників, що містять позитивні та негативні слова та їхні ваги.

– Використання методів машинного навчання, таких як класифікація, нейронні мережі або глибоке навчання, для визначення тональності тексту.

– Врахування контексту та нюансів, які можуть впливати на оцінку тональності.

3. Визначення показників та візуалізація результатів:

3.1. Визначення показників:

– Розрахунок загальної тональності відгуку на основі аналізу окремих слів та їхніх ваг.

– Визначення категорій тональності, таких як позитивна, негативна або нейтральна.

– Створення числових метрик, які відображають рівень тональності (наприклад, на шкалі від -1 до +1).

3.2. Візуалізація результатів:

– Відображення результатів аналізу у зручному для сприйняття вигляді, такому як графіки, діаграми чи таблиці.

– Відмітка позитивних та негативних аспектів відгуку для більш детального аналізу.

Діаграма IDEF0 (Integration Definition for Function Modeling) "ЯК БУДЕ" – це графічний інструмент, що використовується для моделювання функцій та їх взаємозв'язків у процесах або системах. Для створення діаграми "ЯК БУДЕ" використовуються блоки, стрілки та тексти для відображення функцій, вхідних та вихідних даних, управління та залежностей між ними (рис. 2.6).

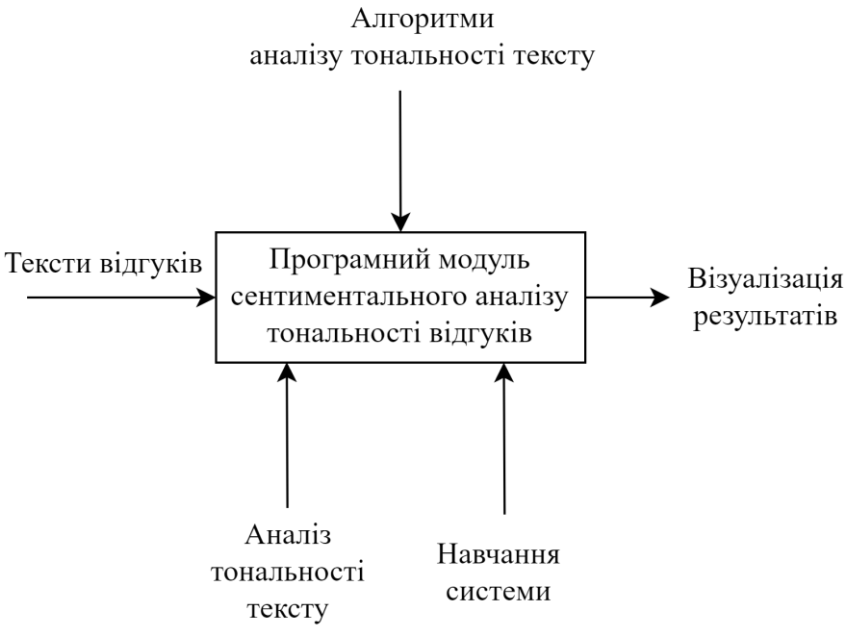


Рисунок 2.6 – Діаграма «ЯК БУДЕ» для програмного модуля сентиментального аналізу тональності відгуків

Описання діаграми IDEF0 "ЯК БУДЕ":

1. Блоки (функції):

– вхідні дані (дані, які вводяться у систему для аналізу);

– аналіз тональності тексту (функція, що виконує аналіз тональності вхідних даних);

– визначення показників (функція, що розраховує числові метрики та результати аналізу);

– візуалізація результатів (функція, що створює графічне відображення результатів);

– вихідні дані (дані, які виводяться з системи).

2. Стрілки (зв'язки):

– від вхідних даних до програмного модуля сентиментального аналізу тональності відгуків (вхідні дані передаються у функцію для подальшого аналізу).

– від функції "Аналіз тональності тексту" до до програмного модуля сентиментального аналізу тональності відгуків (результати аналізу передаються для розрахунку метрик).

– від функції до програмного модуля сентиментального аналізу тональності відгуків до функції "Візуалізація результатів" (метрики передаються для створення графічного відображення).

Дані для діаграми IDEF0 "ЯК БУДЕ" можуть бути більш деталізовані або включати додаткові функції та зв'язки, залежно від потреб конкретної системи чи процесу, який моделюється.

Декомпозиція процесів в форматі IDEF0 забезпечує більш детальне подання функцій та їх взаємозв'язків. Нижче наведена декомпозиція процесів, описаних вище, в форматі IDEF0:

1. Функція: Вхідні дані

1.1. Прийняття вхідних даних

1.2. Перевірка формату та достовірності вхідних даних

2. Функція: Аналіз тональності тексту

2.1. Токенізація тексту

2.2. Лематизація або стемінг слів

2.3. Видалення стоп-слів

2.4. Використання словникових або машинного навчання для визначення тональності

2.5. Врахування контексту та нюансів

3. Функція: Визначення показників

3.1. Розрахунок загальної тональності

3.2. Визначення категорій тональності

3.3. Створення числових метрик

4. Функція: Візуалізація результатів

4.1. Підготовка даних для графічного відображення

4.2. Створення графічного відображення результатів

4.3. Передача графічного відображення для подальшого використання

5. Функція: Вихідні дані

5.1. Форматування вихідних даних

5.2. Виведення вихідних даних

Декомпозиція процесів у форматі IDEF0 дозволяє подробішо розглянути кожну функцію та встановити більш чіткі зв'язки між ними. Вона також дозволяє здійснити подальшу деталізацію кожної функції, якщо це необхідно.

Висновки до розділу 2

В розділі було проаналізовано проблематику задачі визначення тональності тексту та проведено аналіз існуючих датасетів.

Для проведення дослідження буде використовуватися модель *Google's BERT*.

BERT – це метод попередньої підготовки мови, що означає, що тренується загальна модель "розуміння мови" на великому текстовому корпусі (наприклад, з Вікіпедії), а потім ця модель використовується для задач обробки природної мови,. *BERT* перевершує попередні методи, оскільки це перша без нагляду, глибоко двонаправлена система для попереднього навчання обробки природної мови [21].

У данному контексті характеристика «Без нагляду означає», що *BERT* навчався, використовуючи лише звичайний текст, що є важливим, оскільки величезна кількість звичайних текстових даних є загальнодоступною в Інтернеті багатьма мовами.

Попередньо навчені представлення також можуть бути контекстно-вільними або контекстуальними, а контекстні подання можуть бути однонаправленими або двонаправленими. Контекстні моделі, такі як *word2vec* або *GloVe*, створюють єдине "вбудовування слова" для кожного слова у словнику. Контекстні моделі замість цього створюють представлення кожного слова, яке базується на інших словах у реченні.

Розділ 3

РОЗРОБКА СИСТЕМИ ВИЗНАЧЕННЯ ТОНАЛЬНОСТІ ТЕКСТІВ УКРАЇНСЬКОЮ МОВОЮ

3.1. Архітектура системи та розгортання необхідних компонентів розробки

Архітектура системи визначення тональності текстів українською мовою включає ряд додаткових модулів, які є необхідними для визначення тональності на рис. 3.1.

Рисунок 3.1 – Архітектура комп’ютерної системи обробки текстової інформації для визначення тональності

Для роботи модуля визначення тональності тексту модулів відносяться:

– модуль визначення мови тексту;

– модуль створення графу сутностей;

– модуль розпізнавання іменованих сутностей;

– модуль визначення тональності тексту.

Модуль розпізнавання мови в контексті аналізу тональності тексту визначає мову, на якій написаний текст. Цей модуль є важливим етапом перед подальшим аналізом тональності, оскільки мова може впливати на правильне розуміння тексту та вибір відповідної моделі аналізу.

Модуль створення графу сутностей відповідає за виявлення та виділення сутностей в тексті, таких як імена людей, організацій, місць тощо. Його завдання полягає у побудові структурованої репрезентації тексту, включаючи зв'язки між різними сутностями. Це допомагає зрозуміти контекст та взаємозв'язки між сутностями в тексті, що може бути корисним при аналізі тональності.

Модуль розпізнавання іменованих сутностей визначає та класифікує іменовані сутності в тексті, такі як імена осіб, організацій, місць тощо. Цей модуль використовує різні методи, включаючи правила, статистику або методи машинного навчання, для розпізнавання цих сутностей та призначення їм відповідних міток або класів. Ця інформація про іменовані сутності може бути використана у модулі визначення тональності для більш детального аналізу тексту та врахування контексту при визначенні тональності.

Ці модулі сприяють покращенню якості та точності визначення тональності тексту, надаючи більш повну та структуровану інформацію про сутності в тексті в межах визначеної мови.

Модуль визначення тональності тексту: Цей модуль використовується для аналізу тональності тексту та визначення, чи є він позитивним, негативним або нейтральним. Він використовує методи машинного навчання, аналізу природної мови або статистики, щоб визначити настрій або емоційний відтінок тексту.

Модуль формування web-сторінки не є прямо пов'язаним з аналізом тональності тексту, але через нього проходить представлення результатів аналізу у веб-інтерфейсі та взаємодія з користувачем.

Модуль формування web-сторінки використовується для створення та відображення web-сторінок з результатами аналізу тональності тексту та введення фраз для перевірки. Цей модуль може бути корисним як для внутрішнього використання системи аналізу, так і для зовнішніх користувачів,

які можуть використовувати веб-сторінку для отримання доступу до результатів аналізу та взаємодії з системою.

Модуль навчання нейронної мережі (який входить в адміністративний модуль) використовується для тренування нейронної мережі, що використовується в системі аналізу тональності тексту. Його головна функція – навчити модель розпізнавати та класифікувати тональність тексту на основі навчальних даних.

Під час навчання, модель аналізує навчальні дані, порівнює отримані результати з очікуваними виходами та вносить корективи до своїх вагових коефіцієнтів, щоб максимізувати точність класифікації.

Модуль навчання нейронної мережі вимагає наявності навчальних даних, які складаються з текстових прикладів з позначеною тональністю. Модель навчається на цих даних, проходячи через ітерації навчання, в яких оновлюються параметри мережі, щоб зменшити помилки в класифікації.

Для розгортання системи встановлюємо середовище розробки:

– *Anaconda*;

– *Anaconda Navigator*.

Anaconda Navigator – це графічний інтерфейс для управління пакетами та середовищами розробки *Python,* який постачається з дистрибутивом *Anaconda*. Це потужний інструмент для налаштування та управління середовищами виконання *Python,* який дозволяє розробникам легко встановлювати, використовувати та управляти різними бібліотеками та пакетами *Python*.

Anaconda Navigator має простий та інтуїтивно зрозумілий інтерфейс користувача, що дозволяє користувачам легко переглядати, встановлювати та управляти пакетами *Python*. Він також має вбудовані інструменти для управління середовищами, такими як створення нових середовищ, встановлення та видалення пакетів з середовищ, та інші.

Крім того, управління середовищами та пакетами виконується в зручному графічному інтерфейсі, що дозволяє користувачам швидко та легко виконувати різноманітні завдання, такі як налаштування проектів, створення віртуальних середовищ та встановлення бібліотек.

Anaconda Navigator також має вбудовані інструменти для роботи з *Jupyter Notebook*, що дозволяє легко створювати та виконувати блокноти з кодом *Python*, ділитися ними з іншими користувачами та вивчати *Python* у зручному середовищі.

Anaconda Navigator – це потужний та зручний інструмент для управління пакетами та середовищами розробки *Python*, що дозволяє користувачам швидко та ефективно налаштовувати та управляти своїми проектами *Python*.

Створюємо нове середовище, для чого потрібно в *Anaconda Navigator* натиснути на клавішу імпорт та вибрати файл з розширенням *.yaml*, щоб встановити потрібні нам бібліотеки (рис. 3.2).

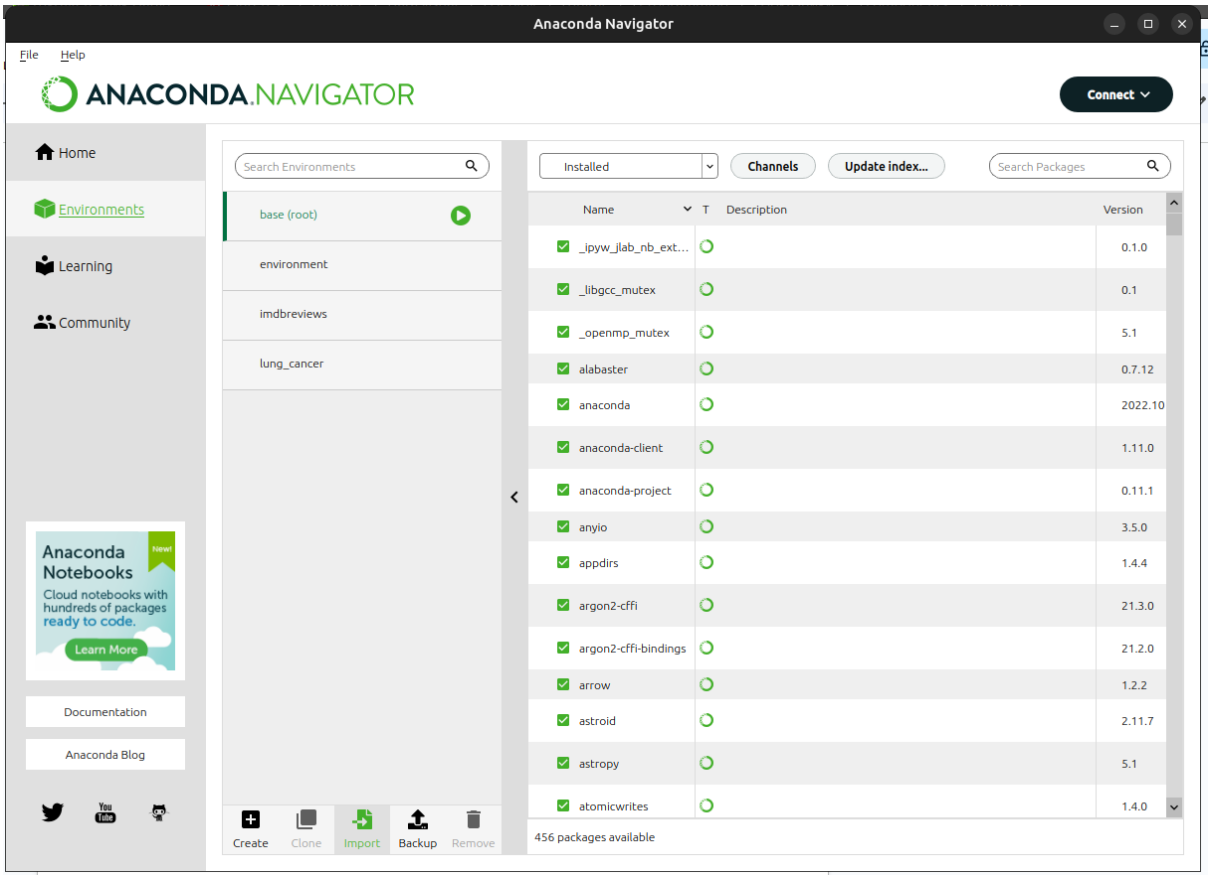


Рисунок 3.2 – Налаштування проєкту в *Anaconda Navigator*

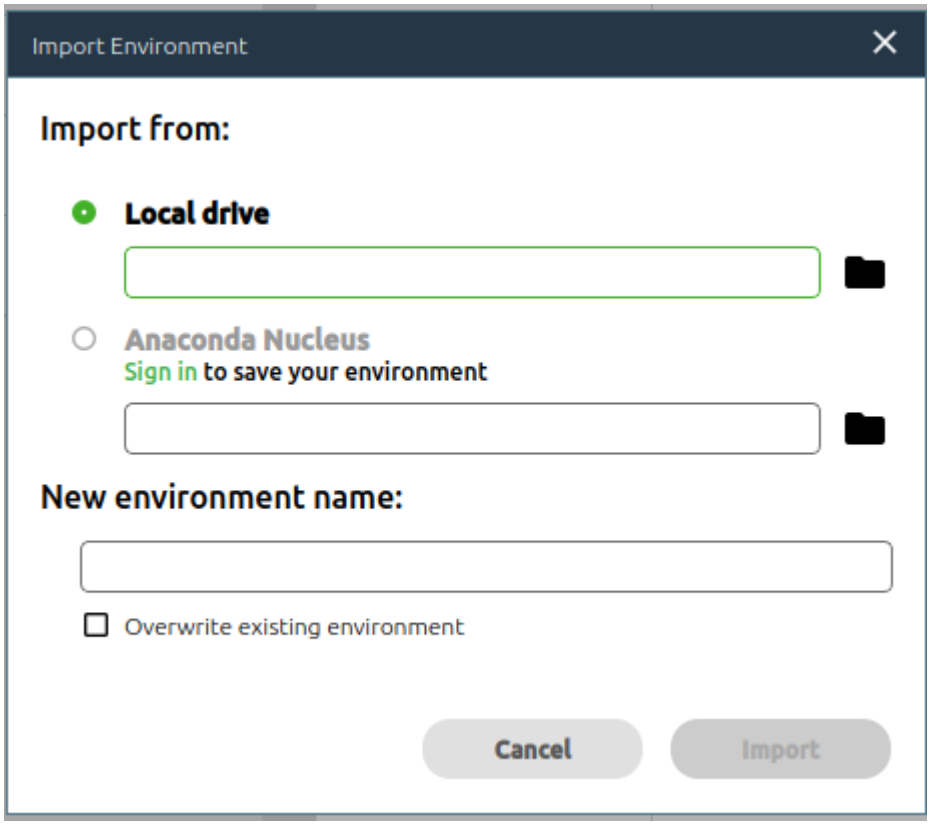


Рисунок 3.3 – Вибір розміщення проєкту в *Anaconda Navigator*

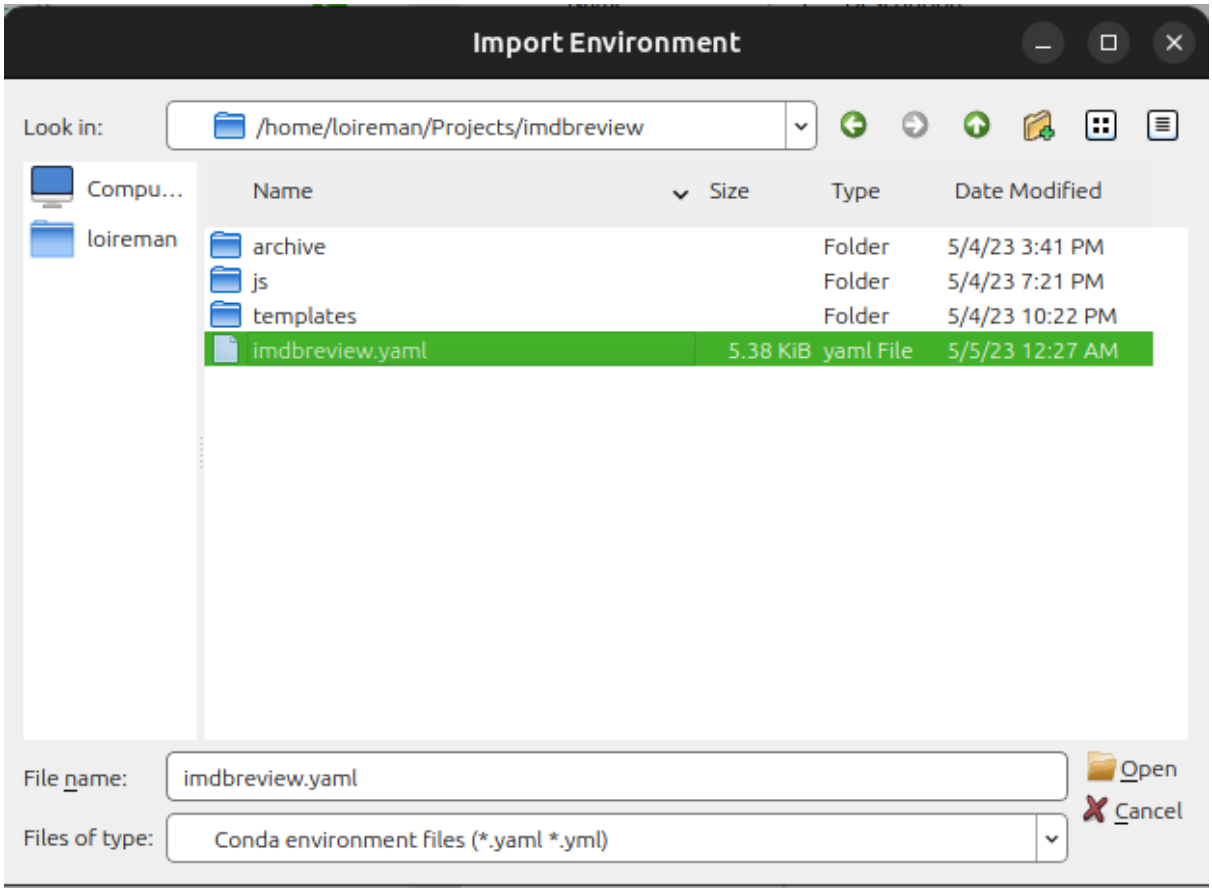


Рисунок 3.4 – Основний модуль проєкту в *Anaconda Navigator*

Після встановлення бібліотек можна користуватись середовищем *Python*.

Пояснення коду, за яким наша модель тренувалась:

*import sqlite*3

import pandas as pd

import sklearn

from sklearn.decomposition import NMF,LatentDirichletAllocation

import numpy as np

import matplotlib

import matplotlib.pyplot as plt

import seaborn as sns

import re

import warnings

warnings.filterwarnings('*ignore*')

print (*pd.__version__*)

print (*sklearn.__version__*)

print (*np.__version__*)

print (*matplotlib.__version__*)

print (*sns.__version__*)

Цей фрагмент коду імпортує бібліотеки *Python,* які використовуються для роботи з базами даних *SQLite,* обробки даних з *Pandas,* машинного навчання з *scikit-learn,* обчислень з *NumPy* та візуалізації даних з *Matplotlib* та *Seaborn*. Також імпортується модуль *re* для роботи з регулярними виразами та модуль *warnings* для управління попередженнями.

У наступному рядку використовується метод *filterwarnings* з модуля *warnings,* щоб ігнорувати всі попередження.

Потім виводяться версії бібліотек *Pandas, scikit-learn, NumPy, Matplotlib* та *Seaborn*.

*con=sqlite*3.*connect*(*r*"*archive/IMDB_Movies*_2021.*db*")

data=pd.read_sql_query("*Select * from REVIEWS*",*con*)

Цей фрагмент коду використовує бібліотеку *sqlite*3 для підключення до бази даних *SQLite* за допомогою методу *connect*. Підключення зберігається у змінній *con*.

Потім використовується метод *read_sql_query* з бібліотеки *Pandas* для виконання *SQL*-запиту до бази даних. Запит вибирає всі дані з таблиці *REVIEWS* та зберігає результат у змінній *data* (*data*.*head*()).

Цей фрагмент коду використовує метод *head* з бібліотеки *Pandas* для виводу перших 5 рядків датафрейму *data*. Це зазвичай робиться для перевірки того, що дані були завантажені правильно та мають очікуваний формат.

```
data.columns
data.drop(['ID','AUTHOR'],inplace=True,axis=1)
map=dict(zip(data.columns.values,data.columns.str.lower().values))
data.rename(columns=map,inplace=True)
data
```

Цей фрагмент коду використовує властивість *columns* з бібліотеки *Pandas* для виводу назв стовпців датафрейму *data*.

Потім використовується метод *drop* для видалення стовпців *ID* та *AUTHOR* з датафрейму. Параметр *inplace=True* означає, що зміни будуть застосовані безпосередньо до датафрейму *data*, а не до його копії. Параметр *axis*=1 означає, що видалення буде проводитись по стовпцях.

У наступному рядку створюється словник *map*, який містить відображення з поточних назв стовпців на їх назви у нижньому регістрі. Це робиться за допомогою методу *zip* та методу *str.lower* з бібліотеки *Pandas*.

Потім використовується метод *rename* для перейменування стовпців датафрейму згідно з словником *map*. Параметр *inplace=True* означає, що зміни будуть застосовані безпосередньо до датафрейму *data*, а не до його копії.

У останньому рядку виводиться датафрейм *data*.

```
data.isna().sum()
ind=data.rating.isna()
```

```
data_na=data.iloc[ind.values,:]
data=data.iloc[~ind.values,:]#clean data
```

Цей фрагмент коду використовує метод *isna* з бібліотеки *Pandas* для перевірки наявності пропущених значень у датафреймі *data*. Потім використовується метод *sum* для підрахунку кількості пропущених значень у кожному стовпці.

У наступному рядку створюється змінна *ind*, яка містить логічний масив, що показує, чи є пропущене значення у стовпці *rating*.

Потім використовується метод *iloc* для створення нового датафрейму *data_na*, який містить тільки ті рядки з датафрейму *data*, у яких є пропущене значення у стовпці *rating*.

У наступному рядку використовується метод *iloc* для заміни датафрейму *data* на його версію без рядків, у яких є пропущене значення у стовпці *rating*. Це робиться за допомогою оператора ~, який інвертує логічний масив *ind.values*.

```
def cleaning(x):
  x=str(x).lower()
  #strip away emojis
  emoji_pattern=re.compile("["
                u"\U0001F600-\U0001F64F"  # emoticons
                u"\U0001F300-\U0001F5FF"  # symbols & pictographs
                u"\U0001F680-\U0001F6FF"  # transport & map symbols
                u"\U0001F1E0-\U0001F1FF"  # flags (iOS)
                u"\U00002702-\U000027B0"
                u"\U000024C2-\U0001F251"
                "]+", flags=re.UNICODE)
  pattern=re.compile(r"[^A-Za-z]+")# strip numbers,symbols
  pattern1=re.compile(r"(?:\A| )[b-df-hj-np-tv-z]{2,}(?: |\Z)")#delete words with only consonants,for example tt or zzz
  pattern2=re.compile(r"\w*([^aeiou])\1{2,}[a-z]*")#delete words with more than 3 same and consecutive consonants
```

```
    pattern3=re.compile(r"\w*([aeiou])\1{2,}[a-z]*")#delete words with more than 3 same and consecutive vowel
    space_pattern=re.compile(" {2,}")#remove consecutives whitespaces char.
    x=emoji_pattern.sub(" ",x)
    x=pattern.sub(" ",x)
    x=pattern1.sub(" ",x)
    x=pattern2.sub(" ",x)
    x=pattern3.sub(" ",x)
    x=x.strip()
    return(x)
data.title=data.title.apply(lambda x: cleaning(x))
data.review=data.review.apply(lambda x: cleaning(x))
```

Цей фрагмент коду визначає функцію *cleaning*, яка приймає один аргумент *x*. Функція призначена для очищення тексту від емодзі, символів, чисел та інших небажаних елементів.

Спочатку аргумент *x* перетворюється у рядок та переводиться у нижній регістр.

Потім створюється регулярний вираз *emoji_pattern*, який використовується для пошуку емодзі у тексті. Цей вираз містить декілька діапазонів *Unicode*-символів, які включають емодзі різних типів.

Також створюються регулярні вирази *pattern*, *pattern*1, *pattern*2 та *pattern*3 для пошуку чисел, символів, слів з тільки приголосними буквами, слів з більше ніж 3 однаковими та послідовними приголосними буквами та слів з більше ніж 3 однаковими та послідовними голосними буквами відповідно.

У наступних рядках коду використовуються методи *sub* з модуля *re* для заміни знайдених елементів на пробіли. Наприклад, усі емодзі у тексті будуть замінені на пробіли.

У останньому рядку коду використовується метод *strip* для видалення початкових та кінцевих пробілів у тексті.

Запуск коду проводиться наступними командами:

– файли *.ipynb* – запуск з *python notebook*;

– файли *.py* – запуск з *python* та редагування звичайним редактором коду

Запуск веб-інтерфейсу сайту проводиться за допомогою команди:

наша директорія> *python app.py*

```
(imdbreview) loireman@cobain:~$ cd /home/loireman/Projects/imdbreview/
(imdbreview) loireman@cobain:~/Projects/imdbreview$ python app.py
2023-05-05 07:12:41.459619: I tensorflow/core/platform/cpu_feature_guard.cc:193]
```

Рисунок 3.5 – Команда запуску веб-інтерфейсу

Після цього у нас завантажиться сайт за адресою *http://localhost*:5000

```
Skipping registering GPU devices...
Loaded model from disk
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment.
 Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

Рисунок 3.6 – Формування сайту за визначеною адресою

3.2. Аналіз роботи розробленої системи аналізу тональності тексту

Застосунок при запуску має декілька параметрів: поле вводу тексту, та кнопка “Завантажити” (рис. 3.7).

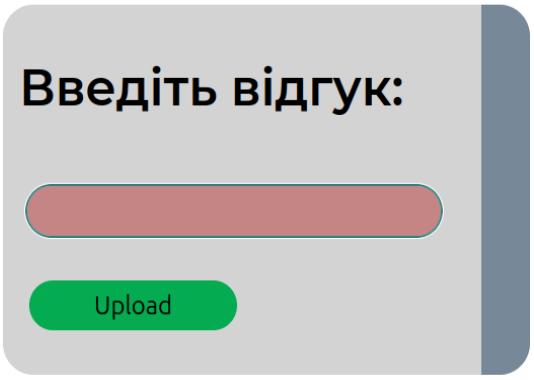


Рисунок 3.7 – Інтерфейс розробленої системи

Якщо у поле (позначене червоним) не ввести значення та натиснути завантажити – виникне наступне повідомлення (рис. 3.8).

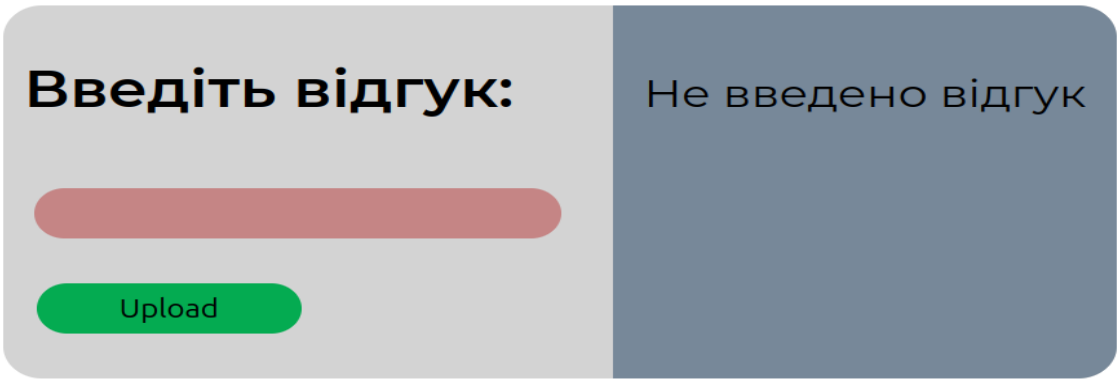


Рисунок 3.8 – Обробка помилок в системі

В іншому випадку, на основі текст відгуку отримаємо оцінку емоційного забарвлення відгуку (погано/задовільно/добре), яку можна замінити на більшу кількість оцінки забарвлення від 1 до 10 (дана властивість налаштовується у коді).

Також будується переклад тексту, що зумовлено тим, що базові набори даних *IMDB* було взято з відкритих джерел і вони були англійською мовою, а вже потім проводилась локалізація (рис. 3.9).

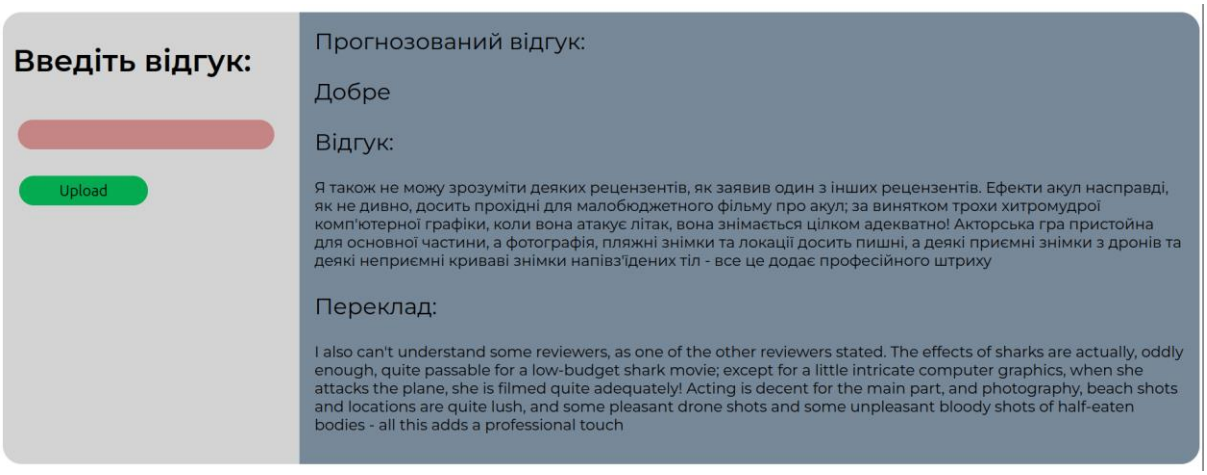


Рисунок 3.9 – Приклад роботи визначення тональності тексту

Наступний фрагмент коду використовує бібліотеки *Matplotlib* та *Seaborn* для створення графіка, який візуалізує відношення тексту до того або іншого тонів.

plt.style.use('*ggplot*')

fig=plt.figure(*figsize*=(15,5))

plt.subplot(121)

sns.countplot(*data.rating,palette=sns.color_palette*('*viridis*'))

plt.ylabel("*Count*")

plt.xlabel("*Rating*")

Спочатку використовується метод *use* з модуля *style* бібліотеки *Matplotlib* для встановлення стилю графіка на *ggplot*.

Потім створюється нове полотно для графіка за допомогою методу *figure* з бібліотеки *Matplotlib*. Параметр *figsize* встановлює розмір полотна у дюймах.

У наступному рядку використовується метод *subplot* для створення нового графіка у полотні. Параметри 121 означають, що графік буде розташований у першому рядку та першому стовпці сітки з одного рядка та двох стовпців.

Потім використовується метод *countplot* з бібліотеки *Seaborn* для створення графіка кількості елементів у кожному класі стовпця *rating* датафрейму *data*. Параметр *palette* встановлює колірну палітру графіка.

У наступних двох рядках використовуються методи *ylabel* та *xlabel* з бібліотеки *Matplotlib* для встановлення підписів до осей *Y* та *X* відповідно.

Побудований графік представлено на рис. 3.10.

Наступний фрагмент коду використовує функцію *print* для виводу відсотка з рейтингом 1 у датафреймі *data*. Спочатку використовується логічне індексування для вибору тільки тих рядків з датафрейму *data*, у яких значення стовпця *rating* дорівнює 1. Потім використовується метод *len* для підрахунку кількості таких рядків. У наступному рядку використовується метод *len* для підрахунку загальної кількості рядків у датафреймі *data*. Потім обчислюється відсоток фільмів з рейтингом 1 шляхом ділення кількості фільмів з рейтингом 1 на загальну кількість фільмів.

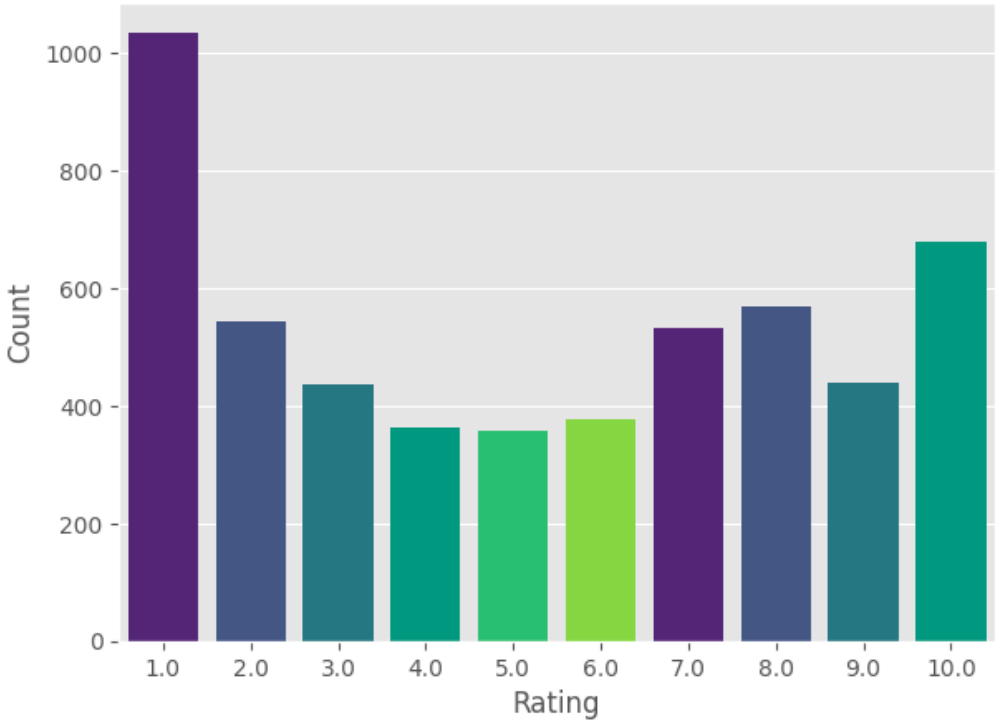


Рисунок 3.10 – Відношення тексту то шкали тональності

У наступному рядку використовується функція *round* для округлення обчисленого відсотка до одного знака після коми. Потім результат виводиться за допомогою функції *print*.

print("*Percentage of Movies with rating* 1: ",*round*(*len*(*data*[*data.rating*==1])/*len*(*data*)*100,1),"%")

Наступний фрагмент коду використовує бібліотеки *Matplotlib* та *Seaborn* для створення гістограм.

plt.figure(*figsize*=(15,7))

plt.subplot(121)

Bad_movies_words=*data*[*data.rating*<5].*review.str.split*().*apply*(*lambda x*:*len*(*x*))

plt.hist(*Bad_movies_words*,*bins*=20,*color*="*darkred*",*label*="*Bad Reviews*")

plt.xlabel("*Number of words*")

plt.legend(*labelcolor*='*black*',*facecolor*="*white*")

plt.subplot(122)

good_movies_words=*data*[*data.rating*>5].*review.str.split*().*apply*(*lambda x*:*len*(*x*))

plt.hist(*Bad_movies_words*,*bins*=20,*color*="*firebrick*",*label*="*Good Reviews*")

plt.xlabel("*Number of words*")

plt.legend(*labelcolor='black',facecolor="white*")

Спочатку створюється нове полотно для графіка за допомогою методу *figure* з бібліотеки *Matplotlib*. Параметр *figsize* встановлює розмір полотна у дюймах.

У наступному рядку використовується метод *subplot* для створення нового графіка у полотні. Параметри 121 означають, що графік буде розташований у першому рядку та першому стовпці сітки з одного рядка та двох стовпців.

Потім використовується логічне індексування для вибору тільки тих рядків з датафрейму *data*, у яких значення стовпця *rating* менше 5. Потім використовується метод *str.split* для розділення тексту у стовпці *review* на слова. Потім використовується метод *apply* для застосування функції *lambda*, яка підраховує кількість слів у кожному рядку. Результат зберігається у змінній *Bad_movies_words*.

У наступному рядку використовується метод *hist* з бібліотеки *Matplotlib* для створення гістограми кількості слів у поганих відгуках. Параметр *bins* встановлює кількість стовпців гістограми, а параметр *color* встановлює колір стовпців. Параметр *label* встановлює підпис до графіка.

У наступних двох рядках використовуються методи *xlabel* та *legend* з бібліотеки *Matplotlib* для встановлення підпису до осі *X* та легенди графіка відповідно.

У наступному рядку використовується метод *subplot* для створення нового графіка у полотні. Параметри 122 означають, що графік буде розташований у першому рядку та другому стовпці сітки з одного рядка та двох стовпців.

Потім використовується логічне індексування для вибору тільки тих рядків з датафрейму *data*, у яких значення стовпця *rating* більше 5. Потім використовуються такі ж методи, що й у попередньому абзаці, для підрахунку кількості слів у хороших відгуках. Результат зберігається у змінній *good_movies_words*.

Змінна *Bad_movies_words*, яка була створена у попередньому фрагменті коду, містить кількість слів у поганих відгуках з датафрейму *data*.

Наступний фрагмент коду визначає функцію *Significant_words,* яка приймає шість аргументів: *Vectorizer*, *Data, ax, title, rotation* та *orientation*. Функція призначена для візуалізації 15 найбільш значимих слів у текстових даних.

```
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
def
Significant_words(Vectorizer,Data,ax=None,title=None,rotation=45,orientation='v'):
  Counts=Vectorizer.transform(Data)
  Counts=Counts.toarray()
  Counts=Counts.sum(axis=0)
  ind=np.argsort(Counts)[-15:]
  if ax==None:fig=plt.figure(figsize=(10,5))
  if orientation=='h':

sns.barplot(y=np.array(Vectorizer.get_feature_names())[ind],x=Counts[ind],ax=ax,o
rient=orientation)
  else:

sns.barplot(x=np.array(Vectorizer.get_feature_names())[ind],y=Counts[ind],ax=ax)
  if title:
     ax.set_title(label=title,loc="center",fontfamily='serif',color="darkblue",size=16)
  ax.set_ylabel("Count")
  ax.set_xlabel("Words")
  ax.tick_params(axis='x',rotation=rotation,color="black")
```

Спочатку імпортуються класи *TfidfVectorizer* та *CountVectorizer* з бібліотеки *scikit-learn*. Ці класи використовуються для векторизації текстових даних.

У наступному рядку використовується метод *transform* з аргументом *Vectorizer* для перетворення текстових даних у матрицю кількостей. Результат зберігається у змінній *Counts*.

Потім використовується метод *toarray* для перетворення матриці кількостей у масив *NumPy*. Потім використовується метод *sum* з параметром *axis*=0 для підрахунку загальної кількості кожного слова у текстових даних.

У наступному рядку використовується метод *argsort* з бібліотеки *NumPy* для сортування масиву кількостей слів за зростанням. Потім використовується індексування за допомогою оператора [-15:] для вибору 15 найбільших значень. Індекси цих значень зберігаються у змінній *ind*.

У наступному рядку перевіряється, чи аргумент *ax* є рівним *None*. Якщо це так, то створюється нове полотно для графîка за допомогою методу *figure* з бîблîотеки *Matplotlib*. Параметр *figsize* встановлює розмîр полотна у дюймах.

Потîм перевîряється значення аргумента *orientation*. Якщо воно рîвне '*h*', то використовується метод *barplot* з бïблïотеки *Seaborn* для створення горизонтальної гîстограми найбîльш значимих слîв. Якщо значення аргумента *orientation* не рîвне '*h*', то використовується метод *barplot* для створення вертикальної гîстограми.

У наступних рядках використовуються методи *set_title, set_ylabel, set_xlabel* та *tick_params* з бïблïотеки *Matplotlib* для встановлення пîдпису до графîка, пîдписîв до осей *Y* та *X* та параметрîв мîток осей вîдповîдно.

Наступний фрагмент коду створює об’єкт *CountVectorizer* з бібліотеки *scikit-learn* та використовує його для векторизації текстових даних.

```
vec_title=CountVectorizer(stop_words='english',min_df=2,max_df=300)
vec_title.fit(data.title)
print(f"Unique unigrams(words) on Titles {len(vec_title.vocabulary_)}")
```

Спочатку створюється об’єкт *vec_title* класу *CountVectorizer*. Параметр *stop_words* встановлює список стоп-слів, які будуть ігноруватися під час векторизації. Параметр *min_df* встановлює мінімальну кількість документів, у яких повинно зустрічатися слово, щоб воно було включене до словника. Параметр *max_df* встановлює максимальну кількість документів, у яких може зустрічатися слово, щоб воно було включене до словника.

У наступному рядку використовується метод *fit* для навчання об’єкта *vec_title* на текстових даних з стовпця *title* датафрейму *data*.

У останньому рядку використовується функція *print* для виводу кількості унікальних уніграм (слів) у заголовках. Це значення обчислюється за допомогою властивості *vocabulary_* об’єкта *vec_title*, яка містить словник уніграм.

Наступний фрагмент коду використовує бібліотеку *Matplotlib* для створення полотна для графіків та функцію *Significant_words*, яка була визначена раніше, для візуалізації найбільш значимих слів у текстових даних.

```
fig=plt.figure(figsize=(20,7))
ax=fig.subplots(1,3)
ax=ax.ravel()
Significant_words(vec_title,data.title,title="Summary of reviews",ax=ax[0])
Significant_words(vec_title,data[data.rating<5].title,title="Negatives summary of reviews",ax=ax[1])
Significant_words(vec_title,data[data.rating>5].title,title="Positives summary of reviews",ax=ax[2])
```

Спочатку створюється нове полотно для графіка за допомогою методу *figure* з бібліотеки *Matplotlib*. Параметр *figsize* встановлює розмір полотна у дюймах.

У наступному рядку використовується метод *subplots* для створення сітки з трьох графіків у полотні. Параметри 1,3 означають, що сітка буде мати один рядок та три стовпці. Результат зберігається у змінній *ax*.

Потім використовується метод *ravel* для перетворення масиву *ax* у одновимірний масив.

У наступних трьох рядках використовується функція *Significant_words* для візуалізації 15 найбільш значимих слів у заголовках всіх відгуків, поганих відгуків та хороших відгуків відповідно (рис. 3.11). Аргументи *vec_title, data.title, data*[*data.rating*<5].*title* та *data*[*data.rating*>5].*title* передаються як аргументи *Vectorizer* та *Data* функції *Significant_words*. Аргументи *title* передаються як аргументи *title* функції *Significant_words*. Аргументи *ax*[0], *ax*[1] та *ax*[2] передаються як аргументи *ax* функції *Significant_words*.

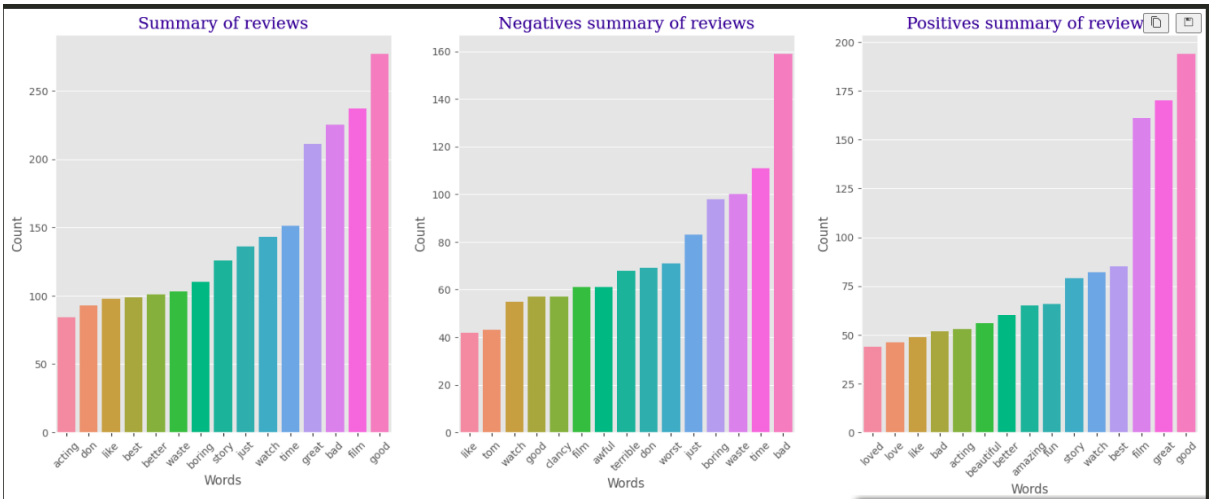


Рисунок 3.11 – Приклад побудованих графіків для візуалізації визначення тональності тексту

3.3. Описання процесу навчання моделі

Для навчання системи використовувалась модель *Kernas*. Наступний фрагмент коду створює словник *class_models*, який містить п’ять об’єктів класифікаторів з бібліотеки *sklearn*: *MultinomialNB*, *SVC*, *ComplementNB*, *Perceptron* та *KNeighborsClassifier*. Кожен з цих класифікаторів ініціалізується з параметрами за замовчуванням. Ці класифікатори можуть бути використані для створення моделей машинного навчання для класифікації.

```
class_models={
        "Naive Bayes":MultinomialNB(),
        "SVC":SVC(),
        "Complement Naive Bayes":ComplementNB(),
        "Perceptron": Perceptron(),
        "Kneighbors": KNeighborsClassifier()
        }
```

Наступний фрагмент коду визначає дві функції: *accurac_model* та *report*. Функція *accurac_model* приймає чотири аргументи: *x*, *y*, *models* та *not_sparse*. Ця функція обчислює середнє значення перехресної перевірки для кожного з моделей у словнику *models*, використовуючи дані у словнику *x* та мітки у масиві *y*.

Параметр *not_sparse* вказує, чи потрібно перетворити розріджені матриці у масиви перед передачею їх у функцію *cross_val_score*. Функція повертає словник з результатами для кожного з наборів даних у словнику *x*.

Функція *report* приймає один аргумент: *scores*. Ця функція виводить на екран результати, збережені у словнику *scores*.

```
def accurac_model(x,y,models,not_sparse=False):
  scores={"Titles":[],"Reviews":[]}
  for j,vec_text in x.items():
     for name,model in models.items():
        if not_sparse:

cr=cross_val_score(model,vec_text.toarray(),y,scoring="neg_mean_squared_error",
)
        else:cr=cross_val_score(model,vec_text,y)
        scores[j].append({name:cr.mean()})
  return scores
def report(scores):
  for name,scores in scores.items():
     print(name,":",)
     for score in scores:
        for model,cv_score in score.items():
           print(f"{model} : {cv_score}")
     print("-"*30)
```

Цей фрагмент коду створює словник *regr_models*, який містить чотири об’єкти регресорів з бібліотеки *sklearn*: *LinearRegression, Ridge, BayesianRidge* та *DecisionTreeRegressor*. Кожен з цих регресорів ініціалізується з параметрами за замовчуванням, крім *DecisionTreeRegressor*, який ініціалізується з параметром *min_samples_split*=50. Ці регресори можуть бути використані для створення моделей машинного навчання для регресії.

```
regr_models={"Linear Regression":LinearRegression(),
```

"*Ridge*":*Ridge*(),

"*Bayesian ridge*": *BayesianRidge*(),

"*Tree*" : *DecisionTreeRegressor*(*min_samples_split*=50)

}

scores=accurac_model(*vocab,data.rating,regr_models,not_sparse=True*)

report(*scores*)

Наступним кроком є виклик функції *accurac_model* з аргументами *vocab, data.rating, regr_models* та *not_sparse=True*. Це обчислює середнє значення перехресної перевірки для кожного з регресорів у словнику *regr_models,* використовуючи дані у словнику *vocab* та мітки у стовпці *rating* датафрейму *data*. Результати зберігаються у словнику *scores*.

Нарешті, викликається функція *report* з аргументом *scores*. Це виводить на екран результати перехресної перевірки для кожного з регресорів.

Наступний фрагмент коду імпортує ряд класів та функцій з бібліотек *keras* та *tensorflow,* а також функцію *train_test_split* з бібліотеки *sklearn*. Ці класи та функції можуть бути використані для створення та навчання нейронних мереж з використанням бібліотек *keras* та *tensorflow*. Функція *train_test_split* може бути використана для розділення даних на навчальну та тестову вибірки.

import keras

from keras.callbacks import EarlyStopping,ReduceLROnPlateau

from keras.models import Sequential

from keras.layers import Dense,Embedding,LSTM,Dropout,Bidirectional

from keras.optimizers import Adam

from keras.preprocessing import text

from sklearn.model_selection import train_test_split

import tensorflow as tf

Наступний фрагмент коду створює два об’єкти зворотного виклику *keras*: *EarlyStopping* та *ReduceLROnPlateau*. Об’єкт *EarlyStopping* ініціалізується з параметрами *monitor='val_loss', patience*=6 та *restore_best_weights=True,* що означає, що навчання буде зупинено, якщо значення функції втрат на

валідаційній вибірці не покращується протягом 6 епох, і кращі ваги будуть відновлені. Об’єкт *ReduceLROnPlateau* ініціалізується з параметрами *monitor*='*val_loss*', *factor*=0.1 та *patince*=3, що означає, що швидкість навчання буде зменшена на 10 разів, якщо значення функції втрат на валідаційній вибірці не покращується протягом 3 епох.

Early_callback=EarlyStopping(*monitor*="*val_loss*",*patience*=6,*restore_best_weights =True*)

Learning_callback=ReduceLROnPlateau(*monitor*="*val_loss*",*factor*=0.1,*patince*=3)

Для підготовки даних для навчання нейронної мережі наступний фрагмент коду імпортує функцію *pad_sequences* з бібліотеки *keras* та використовує її. Спочатку визначається максимальна кількість ознак як довжина списку імен ознак, повернутого методом *get_feature_names* об’єкта *vec_review*. Максимальна довжина послідовностей встановлюється рівною 600. Дані у стовпцях *review* та *rating* датафрейму *data* розділяються на навчальну та тестову вибірки з використанням функції *train_test_split* з параметром *test_size*=0.1.

from keras.utils.data_utils import pad_sequences

max_features=len(*vec_review.get_feature_names*())

max_length=600

x_train,x_test,y_train,y_test =train_test_split(*data.review,data.rating,test_size*=0.1)

tokenizer = text.Tokenizer(*num_words=max_features*)

tokenizer.fit_on_texts(*x_train*)

x_train = tokenizer.texts_to_sequences(*x_train*)

x_train = pad_sequences(*x_train, maxlen=max_length*)

tokenized_test = tokenizer.texts_to_sequences(*x_test*)

X_test = pad_sequences(*tokenized_test, maxlen=max_length*)

Наступним кроком є створення об’єкта *Tokenizer* з ім’ям *tokenizer* з параметром *num_words=max_features*. Метод *fit_on_texts* застосовується до навчальних даних для створення словника. Метод *texts_to_sequences* застосовується до навчальних та тестових даних для перетворення тексту у послідовності цілих чисел. Нарешті,

функція *pad_sequences* застосовується до навчальних та тестових послідовностей з параметром *maxlen=max_length* для приведення їх до однакової довжини.

Для створення послідовної нейронної мережі з використанням класу *Sequential* з бібліотеки *keras* використовується наступний код:

```
model=Sequential()
model.add(Embedding(input_dim=max_features,output_dim=100,input_length=max_length))
model.add(Dropout(0.5))
model.add(Bidirectional(LSTM(units=124 , return_sequences = True )))
model.add(Bidirectional(LSTM(units=62 )))
model.add(Dropout(0.2))
model.add(Dense(units = 100, activation = 'relu'))
model.add(Dropout(0.2))
model.add(Dense(units = 10, activation = 'relu'))
model.add(Dense(1, activation=None))
```

Мережа складається з декількох шарів: шару *Embedding*, двох шарів *Dropout*, двох шарів *Bidirectional LSTM*, двох шарів *Dense* та вихідного шару *Dense* з одним нейроном та без функції активації. Параметри цих шарів встановлюються відповідно до значень змінних *max_features* та *max_length*, а також деяких інших значень. Ця мережа може бути скомпільована та навчена для розв'язання задачі регресії.

Для компіляції та навчання нейронної мережі використовуються наступні рядки.

```
optimizer=Adam(learning_rate=0.001)
model.compile(optimizer=optimizer,loss="mse")
model.fit(x_train,y_train,epochs=20,validation_data=(X_test,y_test),batch_size=100,callbacks=[Early_callback,Learning_callback])
```

Спочатку створюється об'єкт оптимізатора *Adam* з ім'ям *optimizer* та параметром *learning_rate*=0.001. Метод *compile* застосовується до об'єкта *model* з параметрами *optimizer=optimizer* та *loss='mse'*, що означає, що мережа буде

оптимізована з використанням оптимізатора *Adam* та функції втрат середньоквадратичної помилки.

Наступним кроком є виклик методу *fit* з аргументами *x_train, y_train, epochs*=20, *validation_data*=(*X_test,y_test*), *batch_size*=100 та *callbacks*=[*Early_callback,Learning_callback*]. Це запускає процес навчання мережі на навчальних даних протягом 20 епох з розміром пакету 100. Дані для валідації передаються у параметрі *validation_data*, а об'єкти зворотного виклику передаються у параметрі *callbacks*.

Після всіх налаштувань отримано НМ з наступними параметрами:

– модель – tensorflow sequential

– оптимізатор – Adam(learning rate = 0.001);

– показник loss – mse;

– шари (рис. 3.12).

Процес навчання наведено на рис. 3.13.

Фрагмент коду *scores = model.evaluate*(*X_test, y_test, steps*=*len*(*y_test*), *verbose*=1) використовує метод *evaluate* об'єкта *model* для обчислення значення функції втрат на тестовій вибірці. Метод *evaluate* приймає три аргументи: *X_test, y_test* та *steps*=*len*(*y_test*).

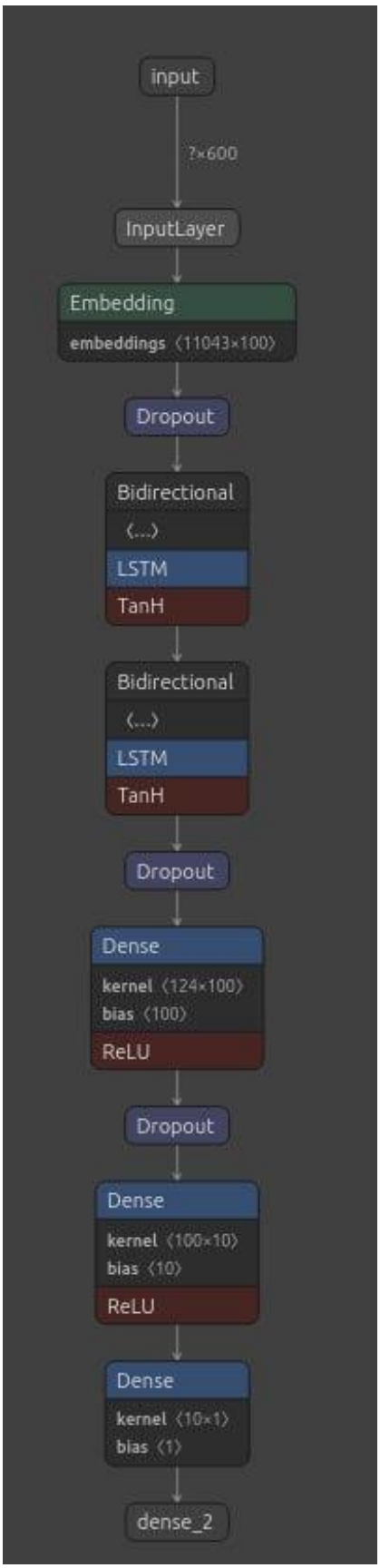


Рисунок 3.12 – Налаштування нейронної мережі для розпізнавання тональності тексту

Це означає, що значення функції втрат буде обчислено на тестових даних з використанням кроку, рівного довжині масиву *y_test*. Параметр *verbose*=1 вказує, що інформація про процес обчислення буде виводитися на екран. Результат обчислення зберігається у змінній *scores*.

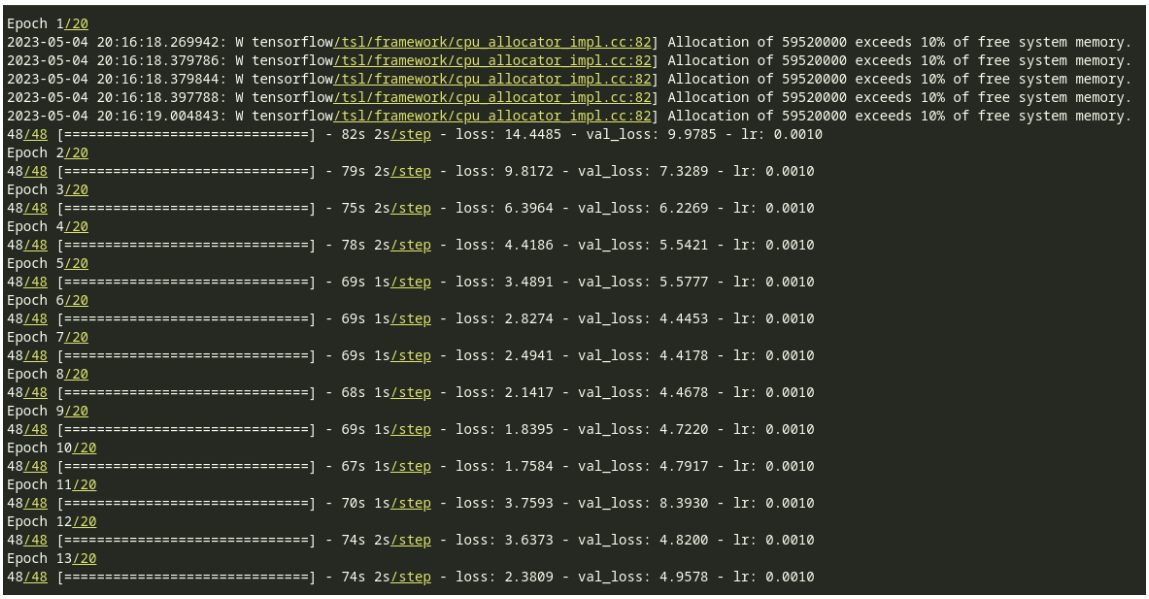


Рисунок 3.13 – Процес навчання нейронної мережі

Наступний фрагмент коду використовує метод *predict* об’єкта *model* для обчислення прогнозованих значень на тестовій вибірці.

534/534 [==============================] – 24*s* 44*ms*/*step* – *loss*: 4.4178

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import ConfusionMatrixDisplay
y_pred = model.predict(X_test)
cm = confusion_matrix(y_test, y_pred.round())
cm_display = ConfusionMatrixDisplay(cm).plot()
```

Результат обчислення зберігається у змінній *y_pred*. Наступним кроком є виклик функції *confusion_matrix* з аргументами *y_test* та *y_pred.round*() для обчислення матриці помилок (рис. 3.13). Результат обчислення зберігається у змінній *cm*. Нарешті, створюється об’єкт *ConfusionMatrixDisplay* з ім’ям *cm_display* та викликається його метод *plot* для відображення матриці помилок.

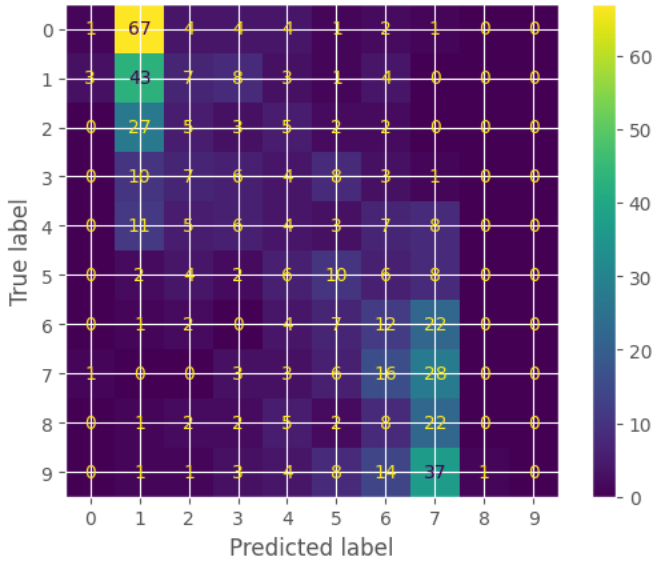
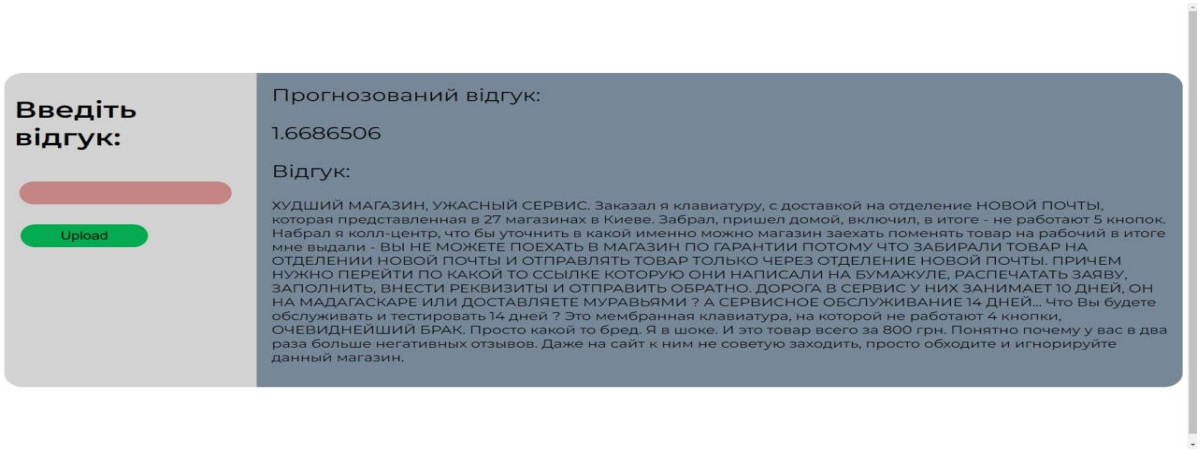


Рисунок 3.14 – Візуалізація матриці помилок

На рис. 3.15 наведено приклади опрацювання відгуків у системі з прямою негативною тональністю.



А)

Б)

Рисунок 3.15 – Приклади відгуку з прямою ознакою негативної тональності: А) великий обсяг тексту, Б) невеликий обсяг тексту

На рис. 3.16 наведено приклад опрацювання відгуку з тональністю, яка прихована у великому тексті.

Рисунок 3.16 – Приклади відгуку з прихованою тональністю

Навчання моделі проводилось на вибірці з 500 відгуків на фільми, які було зібрано з загальнодступних мереж. Експертом з визначення тональності відгуку виступав особисто розробник системи.

За рахунок використання вбудованої системи перекладу також є можливість розпізнавати тональність відгуків не тільки українською мовою (рис. 3.17).

Рисунок 3.17 – Приклади обробки відгуку, що написано іншою мовою

Висновки до розділу 3

В розділі розглянуто архітектуру системи визначення тональності текстів українською мовою, яка складається з компонентів розробки, таких як модуль збору та підготовки даних, модуль векторизації тексту, модуль класифікації тональності, модуль візуалізації результатів.

Проведено аналіз роботи розробленої системи аналізу тональності тексту українською мовою, в якому були проаналізовані результати тестування, визначено ступінь точності та ефективності системи.

Описано процес навчання моделі, у якому розглянуто особливості відбору та підготовки даних для навчання, побудову моделі та її оптимізацію.

На основі розглянутих результатів можна зробити висновок про те, що розроблена система аналізу тональності тексту українською мовою є ефективним інструментом для визначення емоційного відгуку користьувачів. Важливим аспектом є те, що система має потенціал для подальшого розвитку та вдосконалення в контексті обробки природної мови та інформаційних технологій.

Висновки

В кваліфікаційній роботі було проведено аналіз методів визначення тональності текстів українською мовою та розроблено нейронну мережу для обробки довільних текстів відгуків.

У першому розділі проводиться аналіз сучасного стану проблеми оцінки тональності текстів, відбувається постановка задач для дипломної роботи з дослідження сервісів оцінки текстової інформації.

Для дослідження сучасного стану проблеми розглядаються напрями обробки природньої мови та її підвиду оцінки тональності тексту. Проводиться огляд ряду задач, які вирішуються застосуванням сучасних сервісів, побудованих на алгоритмах зазначених вище технологій. На сьогоднішній день, системи активно використовуються для бізнес задач та забезпечують значне підвищення ефективності розвитку бізнесу.

Проводиться огляд основних критичних моментів, які негативно впливають на рівень точності сучасних систем оцінки тональності тексту. У цих напрямках активно проводяться дослідницькі роботи, однак, у повному обсязі дані проблеми досі лишаються не вирішеними.

Було проведено дослідження роботи сервісів оцінки тональності тексту, які представлені на сьогоднішній день широкому загалу для вирішення бізнес-цілей. Це дозволило побачити ефективність представлених систем та оцінити корректність їх роботи у реальних умовах.

В розділі було проаналізовано проблематику задачі визначення тональності тексту та проведено аналіз існуючих датасетів.

Для проведення дослідження використовуватися модель *Google's BERT* та бібліотека *Keras*.

Попередньо навчені представлення були контекстно-вільними, а контекстні подання були двонаправленими. Контекстні моделі дозволили створити єдине "вбудовування слова", що дозволило створити представлення кожного слова, яке базується на інших словах у реченні.

В третьому розділі розглянуто архітектуру системи визначення тональності текстів українською мовою, яка складається з компонентів розробки, таких як модуль збору та підготовки даних, модуль векторизації тексту, модуль класифікації тональності, модуль візуалізації результатів.

Проведено аналіз роботи розробленої системи аналізу тональності тексту українською мовою, в якому були проаналізовані результати тестування, визначено ступінь точності та ефективності системи.

Описано процес навчання моделі, у якому розглянуто особливості відбору та підготовки даних для навчання, побудову моделі та її оптимізацію.

На основі розглянутих результатів можна зробити висновок про те, що розроблена система аналізу тональності тексту українською мовою є ефективним інструментом для визначення емоційного відгуку користьувачів. Важливим аспектом є те, що система має потенціал для подальшого розвитку та вдосконалення в контексті обробки природної мови та інформаційних технологій.

Список використаних джерел

1. *Bing Liu, Sentiment Analysis*: *Mining Opinions, Sentiments, and Emotions – Cambridge University Press*; 1 *edition* – 2015, 383 с.

2. *Cambria, E., Das, D., Bandyopadhyay, A Practical Guide to Sentiment Analysis, Springer* – 2017, 199 с.

3. *The best sentiment analysis tools / TalkWalker, URL*: *https*://*www.talkwalker.com*/*blog*/*best-sentiment-analysis-tools* (Дата звернення: Квітень 2, 2023)

4. *Gerardus Blokdyk, Sentiment Analysis a Complete Guide,* 5*starcooks* – 2018, 126 с.

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: 2023.03.10.

6. *Passenger dragged off overbooked United flight, URL*: *https*://*edition.cnn.com*/2017/04/10/*travel*/*passenger-removed-united-flight-trnd*/*index.html.* Дата звернення: 2023.03.10.

7. *Soudamini Hota, Sudhir Pathak, KNN classifier based approach for multi-class sentiment analysis of twitter data, Independently publisher* – 2017, 124 с.

8. *Sentiment Analysis*: *learn everything you need to know, URL*: *https*://*monkeylearn.com*/*sentiment-analysis*/. Дата звернення: 2023.03.10.

9. *Trump vs Hillary*: *Sentiment analysis on Twitter mentions, URL*: *https*://*monkeylearn.com*/*blog*/*trump-vs-hillary-sentiment-analysis-twitter-mentions*/ . Дата звернення: 2023.03.11.

10. *IMDB Movies Review dataset, URL*: *https*://*www.kaggle.com*/*iarunava*/*imdb-movie-reviews-datasetb*. Дата звернення: 2023.04.02.

11. *Francois Chollet, Deep Learning with Python Languagr,* 1*st Edition* – 2017, 384 *p.*

12. *Pradeep Pujari, Md. Rezaul Karim, Practical Convolutional Neural Networks, Packt Publishing* – 2018, 218 *p.*

13. *Frank Millstein, Deep Learning*: 2 *Manuscripts – Deep Learning With Keras And Convolutional Neural Networks In Python Paperback* – 2018, 260 *p*.

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

15. *Frank Millstein, Python Machine Learning*: *Introduction To Machine Learning With Python, Kindle Edition* – 2018, 134 *p*.

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

17. *Jonathon Chambers, Recurrent Neural Networks for Prediction*: *Learning Algorithms, Architectures and Stability, Leicester* – 2011, 105 *p*.

18. *LazyProgrammer*, *Deep Learning*: *Recurrent Neural Networks in Python*: *LSTM, GRU, and more RNN machine learning architectures in Python and Theano* (*Machine Learning in Python*), *Kindle Edition* – 2016, 56 *p*.

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

20. Аналіз тональності тексту [Електронний ресурс] – Режим доступу до ресурсу: *https://en.wikipedia.org/wiki/Sentiment_analysis*

21. *Natural language processing, Towardsdatascience* – [Електронний ресурс]. – Режим доступа: *https://towardsdatascience.com/your-guide-to-natural-language-processing-nlp-48ea2511f6e1*

22. *Natural language processing, Wikipedia* – [Електронний ресурс]. – Режим доступа: *https://en.wikipedia.org/wiki/Natural_language_processing*

23. *Natural language processing, Towardsdatascience* – [Електронний ресурс]. – Режим доступа: *https://towardsdatascience.com/your-guide-to-natural-language-processing-nlp-48ea2511f6e1*

24. 5 *Use Cases for Natural Language Processing* – [Електронний ресурс]. – Режим доступа: *https://customerthink.com/5-use-cases-for-natural-language-processing-application-in-marketing/*

25. *How to Use Sentiment Analysis to Manage Your Brand Reputation & Win Customers* – [Електронний ресурс]. – Режим доступа: *https://brandmentions.com/blog/sentiment-analysis/*

26. *Sentiment Analysis in Marketing* – [Електронний ресурс]. – Режим доступа: *https://www.cmswire.com/digital-marketing/sentiment-analysis-in-marketing-what-are-you-waiting-for/*

27. *Marketing Research*: *The Role of Sentiment Analysis* – [Електронний ресурс]. – Режим доступа: *http://wps.fep.up.pt/wps/wp489.pdf*

28. *Sentiment Analysis*: *Types, Tools, and Use Cases* – [Електронний ресурс]. – Режим доступа: *https://www.altexsoft.com/blog/business/sentiment-analysis-types-tools-and-use-cases/*

29. *Sentiment Analysis of Product Reviews Cases* – [Електронний ресурс]. – Режим доступа: *https://monkeylearn.com/blog/sentiment-analysis-of-product-reviews/#:~:text=Sentiment%20analysis%20is%20the%20automated,%3A%20Positive%2C%20Neutral%2C%20Negative.&text=Compare%20your%20product%20reviews%20with%20those%20of%20your%20competitors.*

30. *Four Pitfalls of Sentiment Analysis Accuracy* – [Електронний ресурс]. – Режим доступа: *https://www.toptal.com/deep-learning/4-sentiment-analysis-accuracy-traps*

31. *How to*: *Sentiment analysis and Opinion Mining* – [Електронний ресурс]. – Режим доступа: *https://docs.microsoft.com/en-us/azure/cognitive-services/text-analytics/how-tos/text-analytics-how-to-sentiment-analysis?tabs=version-3-1*

32. *Sentiment Analysis* – [Електронний ресурс]. – Режим доступа: *https://www.twinword.com/api/sentiment-analysis.php*

Додаток А

Програмний код формування *html*-сторінки з результатами аналізу тональності тексту

```
<style>
  @import
url('https://fonts.googleapis.com/css2?family=Montserrat:wght@400;600&displa
y=swap');
  body {
    font-family: 'Montserrat', sans-serif;
  }
  .button{
    background-color: #04AB51;
  }
  .button:hover{
    background-color: #037538;
  }
</style>

<div style="
    height: 100vh;
    display: grid;
    align-content: center;
    justify-content: center;
  ">
    <div style="
      border-radius: 30px;
      background-color: lightgrey;
```

```
  overflow: clip;
">
  <div style="
  display: flex;
  gap: 16px;
  ">
    <div style="
      padding: 16px;
    ">
      <h1 style="font-size: 48px;">Введіть відгук:</h1>
      <form method=post>
        <input style="
        background-color: #c58585;
        padding: 10px 24px;
        margin: 32px 8px;
        font-size: 24px;
        color: white;
        border: 0;
        border-radius: 30px;
        " type=text name=text>
        <br>
        <input class="button" style="
        margin: 10px;
        padding: 10px 64px;
        font-size: 24px;
        border: 0;
        border-radius: 30px;
        " type=submit value=Upload>
      </form>
    </div>
```

```
      <div style="
        background-color:lightslategrey;
        padding: 24px;
      ">
        <div>
          <span style="
          font-size: 36px;
          ">
            {% if response==1 %}
            Прогнозований відгук:
            {% endif %}
          </span>
          <p style="font-size:36px">{{ result }}</p>
        </div>

      <div>
        {% if response==1 %}
        <p style="font-size:36px">Відгук:</p>
        <p style="font-size:24px">{{ review }}</p>
        <p style="font-size:36px; display: none;">Переклад:</p>
        <p style="font-size:24px; display: none;">{{ translated }}</p>
        {% endif %}
      </div>
      </div>
    </div>
  </div>
</div>
```