

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____

Кафедра _____ Штучного інтелекту _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 122 Комп'ютерні науки _____
(код і повна назва)

Тип програми _____ освітньо-наукова _____
(освітньо-професійна або освітньо-наукова)

Освітня програма _____ Системи штучного інтелекту _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«_____» _____ 20__ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Онищенко Костянтину Георгійовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Веб-орієнтована програмна система для класифікації емоційної забарвленості тексту

затверджена наказом університету від 21 квітня 2025 р. № 295Ст

2. Термін подання студентом роботи до екзаменаційної комісії 3 червня 2025 р.

3. Вихідні дані до роботи Вимоги до веб-застосунку, що буде розроблено, сервер Flask, Jinja2, мова розробки Python, середовище розробки PyCharm, SQLite, SQLAlchemy, Keras.

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Аналіз предметної галузі та постановка задачі

2) Аналіз існуючих систем та формування вимог до програмної системи

3) Архітектура та проектування програмного забезпечення

4) Програмна реалізація

5) Тестування програмного забезпечення

РЕФЕРАТ

Пояснювальна записка: 57 с., 11 рис., 2 дод., 22 джерела.

ВЕБ-ЗАСТОСУНОК, ВІДГУКИ, ДАНІ, ЕМОЦІЇ, КЛАСИФІКАЦІЯ,
НЕЙРОННА МЕРЕЖА, ТЕКСТ, ТОНАЛЬНІСТЬ.

Об'єкт розробки – веб-застосунок з розпізнавання емоцій за текстом.

Мета розробки – вирішення проблеми великих витрат часу і кадрового ресурсу на отримання зворотного зв'язку від клієнтів і обробки їх реакцій.

Метод рішення – Flask, Jinja2, SQLAlchemy, мова Python, середовище розробки PyCharm, SQLite, Keras.

У результаті роботи виконано розробку веб-застосунку, що дозволяє виконувати розпізнавання емоцій за текстом. А також виконано аналіз та проектування нейронної мережі для класифікації емоцій за текстом.

ABSTRACT

Master's thesis contains: 57 pp., 11 fig., 2 ann., 22 references.

CLASSIFICATION, EMOTIONS, FEEDBACK, DATA, NEURAL NETWORK, TEXT, TONE, WEB APPLICATION.

Object of development – a web application for emotion recognition from text.

Purpose of development – solving the problem of high time and personnel costs for obtaining customer feedback and processing their reactions.

Solution method – Flask, Jinja2, SQLAlchemy, Python language, PyCharm development environment, SQLite, Keras.

As a result of the work, a web application was developed that enables emotion recognition from text. Additionally, analysis and design of a neural network for emotion classification from text were performed.

ЗМІСТ

Вступ.....	7
1 Аналіз предметної галузі та постановка задачі	9
1.1 Аналіз предметної галузі	9
1.2 Постановка задачі	10
1.3 Методи розпізнавання емоційного забарвлення тексту	12
2 Аналіз існуючих систем та формування вимог до програмної системи ..	14
2.1 Аналіз існуючих систем.....	14
2.1.1 Сервіс ParallelDots	14
2.1.2 Сервіс Google Cloud Natural Language	15
2.1.3 Сервіс Tone Analyzer.....	17
2.1.4 Сервіс Lexalytics Semantria	17
2.1.5 Висновки	19
2.2 Вимоги до програмної системи.....	20
3 Архітектура та проектування програмного забезпечення.....	25
3.1 UML проектування ПЗ	25
3.2 Проектування архітектури ПЗ.....	27
3.3 Проектування структури зберігання даних.....	33
3.4 Аналіз існуючих алгоритмів та методів	34
3.5 UI/UX Design.....	36
4 Програмна реалізація	39
5 Тестування програмного забезпечення	47
Висновки.....	50
Перелік джерел посилання	51
Додаток А Лістинг коду	54
Додаток Б Відомість кваліфікаційної роботи	57

ВСТУП

Розвиток комп'ютерних технологій нині відбувається дуже стрімко. Інтернет, соціальні мережі, мобільний зв'язок та інше нині є невід'ємною частиною нашого життя. Соціальні мережі також стрімко розвиваються та підлаштовуються під запити користувачів. Ми можемо спостерігати вражаючі результати точності підбору контенту за нашими вподобаннями, переглядами. Соціальні мережі також розпізнають нашу зацікавленість контентом або навпаки за тим, як швидко ми прогорнули запропоноване відео, тобто чи затрималися ми його подивитися, чи додивилися до кінця.

Компанії, які зараз використовують всі можливі ресурси для підвищення ефективності своєї роботи є успішними та конкурентоспроможними на сучасному ринку. Задля отримання високих показників прибутку та роботи використовується метод автоматизації простих бізнес-процесів. Отже, технології, які спроможні допомогти у вирішенні вищевказаних задач є популярними на ринку.

Неймовірно великі обсяги даних, які створює та використовує людина стали поштовхом для виникнення Big Data Science [1].

Big Data Science (BDS) – це наука, яка займається обробкою, аналізом та вивченням величезних обсягів даних (так званих Big Data), щоб виявити закономірності, тренди, зв'язки та інші корисні знання. Це міждисциплінарна галузь, що поєднує комп'ютерні науки, статистику, математику та бізнес-аналітику.

Все, що використовує людина – це набір даних, які несуть певну інформацію, але ця інформація може приносити користь лише у випадку вмілого її використання та обробки, а отже наука про великі дані є необхідністю для сучасного світу. Саме завдяки BDS стали головним поштовхом для значного прориву у розвитку штучного інтелекту та машинного навчання.

В наших реаліях майже кожен бізнес залежний від відгуків клієнтів. Клієнтів стає все більше, інформації для обробки з метою покращення продуктивності, роботи та ін. теж стає пропорційно більше, а отже виникає пряма необхідність у найманні додаткових фахівців, які б займалися обробкою та аналізом всієї інформації, що надходить від клієнтів.

Необхідно розуміти, що наймання додаткових фахівців несе за собою додаткові витрати, що для бізнесів як великих, так і малих не є бажаним, а отже є потреба в автоматизації процесу, а саме в автоматизації розпізнавання емоційного забарвлення тексту відгуків та коментарів, використавши нейронні мережі для створення online системи.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної галузі

Технології в сучасному світі розвиваються дуже стрімко, а разом із ними змінюється і підхід до ведення бізнесу. Зараз немає компаній, які б не прагнули автоматизувати більшість внутрішніх процесів, що допомогло б їм зменшити витрати на додаткове працевлаштування персоналу, а також автоматизація процесів прискорює сам процес роботи. Наразі складно уявити навіть простий процес «запис до спеціаліста» без додатку, де було б видно вільні дні, години та інше, а не блокнотик із записами, який можна загубити чи зіпсувати.

Тим не менш все одно залишається перелік певних процесів, які не піддаються автоматизації або ж їх складно автоматизувати і це призведе не до зменшення витрат, а навпаки до їхнього збільшення. Це стосується процесів, де необхідні навички обробки та аналізу. Людина спираючись на свої знання та досвід здебільшого досить легко справляється з такою поставленою задачею. До таких задач можна віднести розпізнавання об'єктів на фото чи відео, перевірка музики на плагіат і обробка текстів [2].

В сучасному бізнес процесі обробляється дуже велика кількість вхідної текстової інформації, а саме така як коментарі, відгуки, які необхідно регулярно обробляти задля складання зворотного зв'язку із клієнтами та покращення своїх послуг.

Головною складністю є те, що такі процеси стандартними способами не вийде автоматизувати, адже вказані задачі потребують використання більш складних та передових технологій зі штучного інтелекту. В цьому допоможуть нейронні мережі, які досить наближено імітують роботу людського мозку та їх можна навчати на наборах даних. Нейронні мережі допомагають пришвидшити із високою точністю виконання задач з розпізнавання тексту [3].

Здебільшого зворотній зв'язок бізнес-компанії отримують саме з написаних онлайн відгуків, статей, на форумах та інше. Користувач, який пише відгук вкладає в нього певне емоційне забарвлення, адже він хоче передати свої емоції через текст якими б вони не були позитивними чи негативними або ж нейтральними.

Найбільш складною задачею є розподіл текстів за емоційним забарвленням. Емоційність тексту зазвичай поділяють на три категорії: позитивний, негативний та нейтральний. Здебільшого такої класифікації вже достатньо для отримання статистики відсоткового відношення кількості відгуків щодо роботи компанії та якості її продукту.

1.2 Постановка задачі

У цій роботі необхідно розробити веб-застосунок, який буде здійснювати аналіз емоційного забарвлення текстових даних.

Основна мета проєкту полягає у створенні рішення, яке дозволить скоротити витрати часу та людських ресурсів на збір, аналіз та обробку зворотного зв'язку клієнтів, враховуючи емоційний аспект їхніх відгуків. Веб-застосунок стане інструментом для автоматизації цього процесу.

Для реалізації веб-застосунку слід виконати наступні завдання:

- організувати систему реєстрації, аутентифікації і авторизації користувачів в системі;
- спроектувати і продумати зручний, інтуїтивно-зрозумілий інтерфейс;
- реалізувати функцію завантаження JSON, XML, CSV файлів і вилучення даних;
- реалізувати функцію класифікації текстових даних за емоціями;
- реалізувати функцію агрегації отриманих після класифікації даних;
- формування звіту.

Під час реалізації серверної частини веб-застосунку треба враховувати наступні вимоги:

- серверна частина повинна бути реалізована за допомогою технології Flask на мові Python;

- в якості системи управління базами даних (СУБД) слід вибрати SQLite [9].

В якості методу вирішення завдання автоматизації розпізнавання емоцій за текстом треба обрати сучасні технології в області машинного навчання, запровадивши нейронні мережі. При реалізації нейронної мережі слід виконати наступні завдання:

- для навчання нейронної мережі треба провести збір даних;

- в якості нейронної мережі слід реалізувати мережу з архітектурою, що підходить для роботи з текстовими даними, яка буде здійснювати класифікацію за 5 параметрами;

- при реалізації архітектури в програмному коді слід використовувати фреймворк Keras [10];

- при побудові архітектури слід використовувати технологію обробки текстових даних word embedding.

Основними користувачами веб-застосунку передбачається компанії середнього та великого бізнесу, що займаються продажем товарів та наданням послуг. Завдяки використанню цього інструменту підприємства зможуть скоротити витрати на утримання персоналу, відповідального за аналіз реакції споживачів у мережі.

Основне джерело прибутку веб-застосунку – співпраця з компаніями з галузі продажів. Відсутність рекламного контенту та стабільна технічна підтримка сприятимуть залученню клієнтів, які цінують ефективність управління часом та ресурсами.

1.3 Методи розпізнавання емоційного забарвлення тексту

У світі стрімкого розвитку інформаційних технологій особливого значення набувають методи автоматичного розпізнавання емоційного забарвлення тексту. Цей напрямок, що належить до галузі обробки природної мови (NLP), спрямований на визначення емоційних станів або настроїв, виражених у текстових повідомленнях. Сьогодні існує кілька підходів до вирішення цього завдання, кожен з яких має свої переваги та недоліки.

Одним з найбільш ранніх і традиційних методів є використання емоційних словників або лексиконів. Такі словники містять списки слів, кожне з яких асоціюється з певною емоцією або полярністю (позитивною, негативною, нейтральною). Наприклад, словник WordNet-Affect класифікує слова за емоційними категоріями, такими як радість, страх чи гнів.

Основний принцип роботи лексичних методів полягає у підрахунку кількості слів з певним емоційним значенням у тексті. Якщо позитивних слів більше, текст вважається оптимістичним, і навпаки.

Перевагами такого підходу є простота реалізації, а отже і менше фінансових затрат, а також незалежність від великих обсягів даних для навчання. Але є і недоліки такого підходу, а саме нездатність враховувати контекст, гумор, іронію та інше, а також є проблемою чутливість до синонімів та метафор.

Інший традиційний підхід полягає у створенні набору правил, які дозволяють аналізувати емоційне забарвлення на основі граматики або синтаксису. Наприклад, використання конструкцій на кшталт «дуже щасливий» підсилює позитивний емоційний контекст.

Перевагами такого підходу є прозорість самого алгоритму та чіткість інтерпретацій. До недоліків можна віднести високу складність розробки для складних мовних структур та обмежена гнучкість у випадку неформальних або творчих текстів.

З розвитком штучного інтелекту значно популярнішими стали методи машинного навчання. Основою таких підходів є алгоритми класифікації, такі як SVM (Support Vector Machine), Naive Bayes або Random Forest. Моделі навчаються на розмічених наборах даних, де кожному тексту присвоєно певну емоційну мітку.

Головною перевагою є швидка адаптація до нових даних та більш висока точність у порівнянні зі звичайними лексичними методами. Недоліками можна назвати потребу у великих наборах розмічених даних та залежність від якості попередньої обробки тексту.

Останніми роками домінуючу роль у розпізнаванні емоційного забарвлення тексту відіграють методи глибинного навчання. Зокрема, нейронні мережі типу RNN (Recurrent Neural Networks) та їх удосконалені версії, такі як LSTM (Long Short-Term Memory) та GRU (Gated Recurrent Units), дозволяють враховувати контекст та послідовність слів.

Особливо перспективним є використання трансформерів, таких як BERT (Bidirectional Encoder Representations from Transformers) або GPT (Generative Pre-trained Transformer). Ці моделі навчаються на величезних корпусах текстів і демонструють високу здатність до розуміння контексту.

Переваги це висока точність та здатність враховувати контекст. Також до переваг можна віднести можливість адаптації до складних емоційних конструкцій. До недоліків відносять високу обчислювальну вартість та потребу у великих обсягах даних для навчання.

Оскільки кожен із методів має свої обмеження, сучасні дослідники часто використовують гібридні підходи, поєднуючи лексичні та машинні методи. Наприклад, можна поєднати лексичний аналіз з нейронною мережею для точнішої інтерпретації текстів.

2 АНАЛІЗ ІСНУЮЧИХ СИСТЕМ ТА ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ

2.1 Аналіз існуючих систем

Задача з розпізнавання емоційного забарвлення тексту не є новою, а отже існують вже готові системи для виконання даної задачі. Головною задачею на початку при розробці своєї системи є спочатку детальний аналіз вже існуючих, виявлення існуючих недоліків та проблем та аналіз із визначенням шляхів їхнього вирішення.

2.1.1 Сервіс ParallelDots

Розглянемо більш детально вже існуючий сервіс ParallelDots [5].

ParallelDots – це компанія, яка спеціалізується на розробці рішень з обробки природної мови (NLP) та комп'ютерного зору з використанням штучного інтелекту та глибинного навчання.

Сервіс ParallelDots надає користувачу змогу провести класифікацію тексту за емоціями на декількох мовах. Класифікація відбувається за шістьма категоріями, такими як радість, злість, захоплення, сум, страх та втома. Система показує користувачу у відсотках на скільки його текст містить ті чи інші емоції. На рисунку 2.1 можна побачити роботу системи та виведені результати обробки тексту.

З інтерфейсу сервісу можна побачити, що користувач може обрати зручну для нього мову із запропонованих. Далі є поле для вводу невеликого за обсягом тексту. Та одразу під текстом результат роботи сервісу.

Як результат аналізу наданого тексту можна побачити, що введений текст відображає злість, трохи суму та жаху.

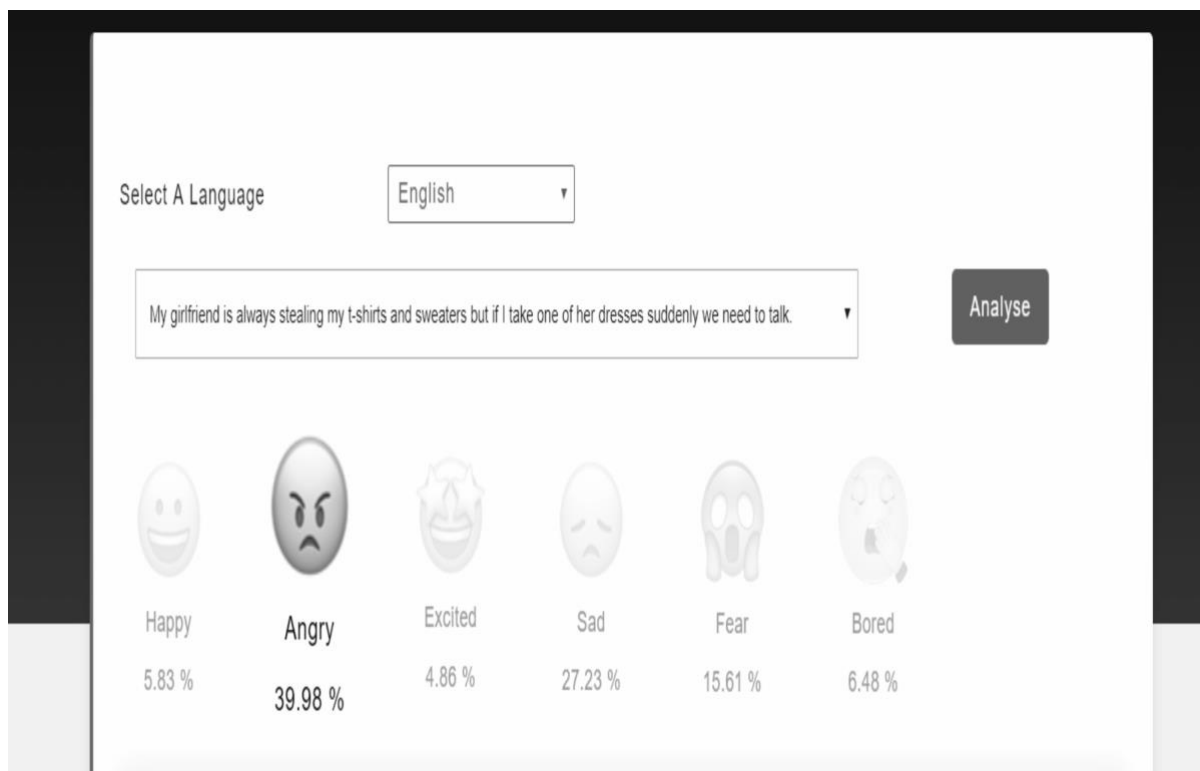


Рисунок 2.1 – Результат роботи сервісу ParallelDots

Даний сервіс можливий для користування лише тим, хто оформив підписку. Із загальнодоступної інформації можна дізнатися, що даний сервіс використовує нейронні мережі при класифікації тексту, але більш детальної інформації не надано.

2.1.2 Сервіс Google Cloud Natural Language

Google Cloud Natural Language – це сервіс від компанії Google, який надає інструменти для обробки природної мови (NLP) за допомогою технологій машинного навчання. Він є частиною платформи Google Cloud Platform (GCP) і дозволяє автоматизувати аналіз тексту, виявлення сутностей, розпізнавання емоційного забарвлення та багато іншого.

Даний сервіс оброблює тексти на 10-тьох мовах. Даний сервіс має більш широкий функціонал, адже як результат ми отримуємо не лише інформацію щодо обробленого тексту, а й розбиті та проаналізовані окремі

його компоненти. Щодо емоційного забарвлення тексту, то сервіс пропонує результати у вигляді позитивних та негативних числових значень. У випадку негативного емоційного забарвлення, значення буде негативне і навпаки. Чим більша величина, тим більш виражена позитивна чи негативна емоція. Недоліком сервісу є те, що він не розмежовує різні емоції, вони поділяються лише на дві категорії: позитивні та негативні (рисунок 2.2).

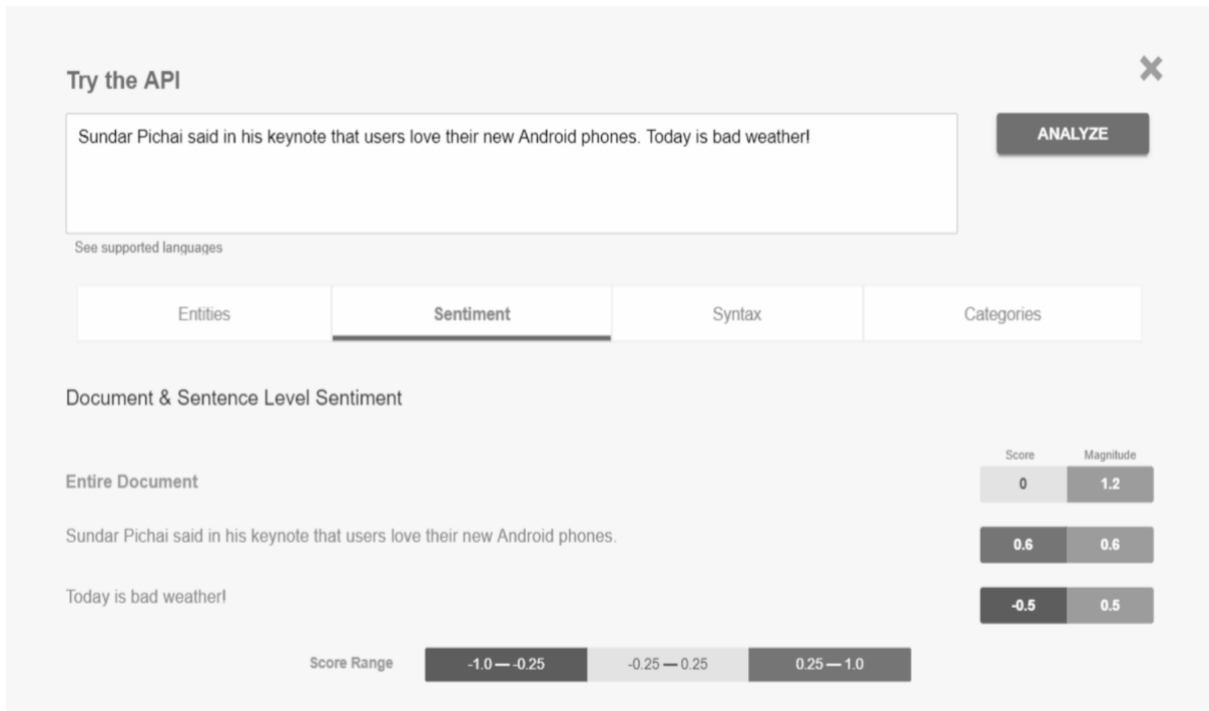


Рисунок 2.2 – Сервіс Google Cloud Natural Language

Даний сервіс використовує нейронну мережу, в тому числі технологію AutoML.

AutoML (Automated Machine Learning) – це технологія, яка автоматизує процес створення, налаштування та оптимізації моделей машинного навчання. Основна мета AutoML – зробити машинне навчання більш доступним для користувачів, які не є експертами у цій галузі, а також спростити процес для досвідчених фахівців.

Розробка моделей машинного навчання традиційно вимагає великих зусиль, включаючи вибір алгоритму, попередню обробку даних, інженерію

ознак, оптимізацію гіперпараметрів та оцінку результатів. AutoML автоматизує більшість цих етапів, що дозволяє отримувати високоякісні моделі з мінімальним втручанням людини.

2.1.3 Сервіс Tone Analyzer

IBM Tone Analyzer – це сервіс від компанії International Business Machines (IBM), який входить до складу платформи IBM Watson. Основне призначення Tone Analyzer – автоматичний аналіз емоційного забарвлення тексту для розуміння настрою, тональності та соціального контексту в письмових комунікаціях [4].

В даному сервісі є поле для вводу тексту, а також є певні налаштування для обробки тексту. Можна обрати тип наданого тексту: це твіт чи імейл, онлайн відгук чи ваш власний текст.

Tone Analyzer класифікує емоцію за 7-ма категоріями: радість, впевненість, нейтральність, сумнів, смуток, страх, злість. Доступ до сервісу. Можуть отримати лише підписники IBM Cloud.

На основі якої нейронної мережі розроблено сервіс та які технології було застосовано не вказано у відкритому доступі.

2.1.4 Сервіс Lexalytics Semantria

Lexalytics Semantria – це платформа для аналізу тексту та обробки природної мови (NLP), розроблена компанією Lexalytics. Semantria орієнтована на бізнес-користувачів і надає потужні інструменти для аналізу великих обсягів тексту, таких як відгуки клієнтів, соціальні медіа, документи та інші джерела даних [8].

Даний сервіс використовує великий набір різноманітних інструментів для обробки тексту. При наданні на обробку неструктурованого тексту, сервіс може виявити іменовані сутності та класифікувати їх за різними

категоріями як місця, організації, імена, час та інше. Також одним із корисних інструментів є можливість створення за текстом анотацій, дайджестів, коротких викладок та інше. При необхідності обробити декілька документів, сервіс може зробити з них короткий звіт.

Сервіс Lexalytics Semantria активно використовує засоби машинного навчання, наприклад для можливості визначити тональність тексту. Детальної інформації щодо засобів, які використовуються немає можливості знайти (рисунок 2.3).

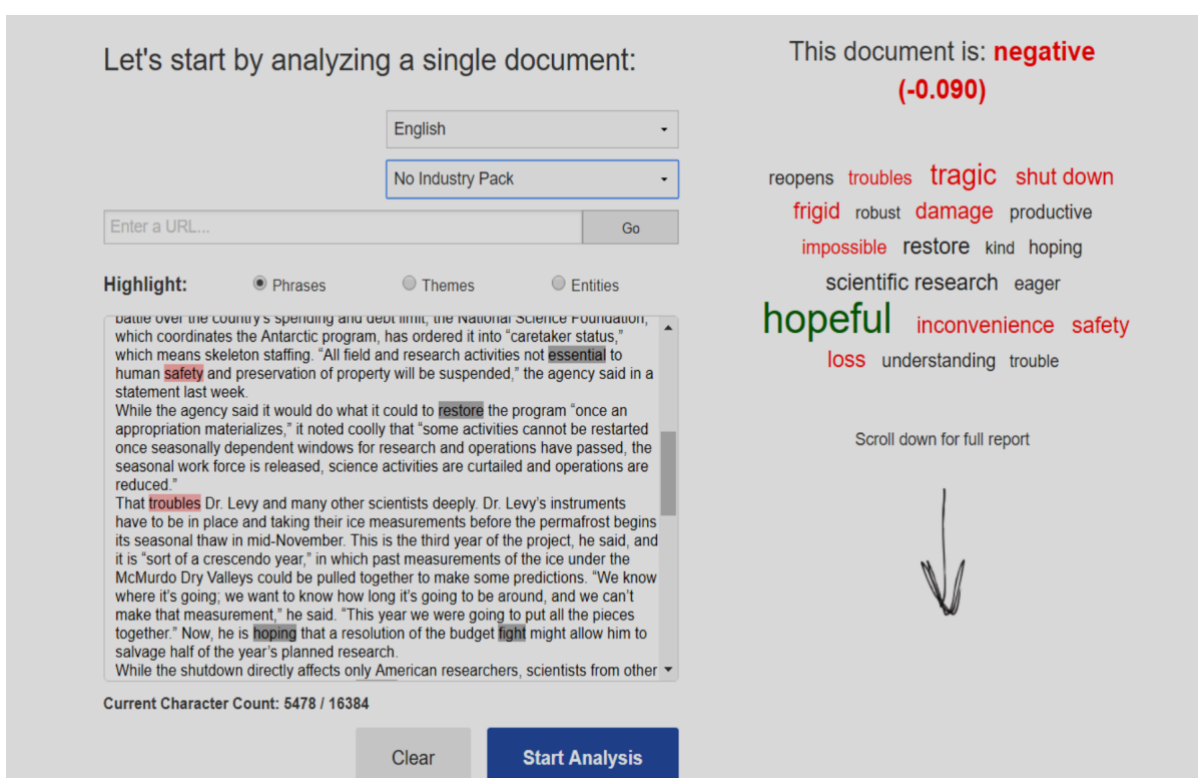


Рисунок 2.3 – Сервіс Lexalytics Semantria

Сервіс має два режими користування: демонстраційний та основний. В демонстраційному режимі користувач більш обмежений у можливостях, наприклад з якою мовою сервіс може працювати, у демонстраційному режимі можна обробити текст на 16 мовах, а в основному на 21 мові. Щоб перейти в основний режим користування необхідно стати підписником.

Даний сервіс розділяє текст за емоційністю лише як позитивний чи негативний, чого для обробки відгуків буде вдосталь.

2.1.5 Висновки

У результаті проведеного аналізу було розглянуто чотири наявні рішення для визначення емоційного забарвлення тексту. Кількість емоційних категорій, на які здійснювалося розбиття, варіювалася від двох до семи.

На основі отриманих результатів було прийнято рішення розробити веб-застосунок для аналізу емоцій на основі текстових даних. Основна мета такого додатку – оптимізувати процес збору зворотного зв'язку від клієнтів та автоматизувати обробку їхніх реакцій за допомогою нейронної мережі. Це дозволить зменшити витрати часу та кадрових ресурсів. Планується застосування сучасних нейромережевих методів, що забезпечать високу доступність сервісу як для індивідуальних користувачів, так і для великих організацій.

Після аналізу наявних аналогів було визначено кілька ключових емоцій. Важливо зазначити, що обмеження емоційного аналізу виключно бінарною класифікацією (позитивне-негативне) призводить до втрати точності та зменшення інформативності результатів. Водночас надмірна кількість емоційних категорій (понад сім) може спричинити перевантаження інформацією, що призведе до складності її інтерпретації та зниження цінності для користувача.

Отже, доцільно обрати обмежену кількість емоційних класів для аналізу, щоб забезпечити баланс між інформативністю та простотою сприйняття результатів. У процесі роботи було обрано чотири основні емоції: радість, смуток, гнів і агресія. Вони є достатньо виразними у текстових повідомленнях, що робить їх оптимальними для аналізу реакцій користувачів.

2.2 Вимоги до програмної системи

Формування вимог є однією з ключових складових процесу розробки будь-якої програмної системи. Вимоги до програмного забезпечення – це сукупність характеристик, функціональних можливостей та якісних параметрів програмного продукту, який знаходиться на стадії проектування або розробки. Вони встановлюються в рамках процесу аналізу вимог і документуються у вигляді специфікацій, діаграм прецедентів та інших артефактів, що створюються під час аналізу та моделювання вимог.

У веб-застосунку повинна бути реалізована підтримка користувачької ролі, оскільки кожен обліковий запис функціонує як відокремлена одиниця. Одна роль є достатньою для забезпечення повноцінної роботи користувача в системі. Доступ до функціоналу системи надається лише авторизованим користувачам, які попередньо пройшли процедуру реєстрації.

Кожна задача, що виконується в системі, являє собою окремий процес аналізу та класифікації текстових даних, пов'язаних з конкретним товаром. Користувач має змогу створювати нові задачі, а також переглядати результати вже виконаних. Для кожного завдання здійснюється обробка текстової інформації, яка стосується обраного об'єкта аналізу. Введення вхідних даних відбувається у визначених форматах – JSON, CSV або XML, що містять масиви текстових відгуків щодо досвіду використання певного продукту.

У результаті виконання завдання користувач отримує агреговану статистичну інформацію, представлену у зручному для сприйняття форматі. Інтерфейс програмної системи повинен відповідати сучасним вимогам дизайну, бути витриманим в єдиному стилі, а також забезпечувати інтуїтивно зрозумілу навігацію та зручність управління для кінцевого користувача.

Одним із ключових критеріїв ефективності роботи даного сервісу є висока точність класифікації емоцій на основі текстових даних. Для

досягнення цієї мети необхідно розробити нейронну мережу зі спеціалізованою архітектурою, адаптованою до обробки тексту, яка на виході формуватиме ймовірнісну оцінку належності тексту до кожного з емоційних класів [11].

Підготовка до навчання моделі передбачає етап збору текстових даних, що будуть використані для процесів тренування та валідації. Для підвищення ефективності навчання необхідно здійснити попередню обробку (препроцесинг) текстової інформації відповідно до характеристик обраного датасету. Зібрану вибірку слід розділити на навчальну підмножину, яка використовується безпосередньо під час навчання, та тестову – призначену для оцінювання здатності моделі до узагальнення, що є важливим чинником підвищення її точності.

Для об'єктивного вимірювання точності класифікації необхідно визначити відповідну метрику, яка буде застосовуватись на етапі тестування моделі.

Веб-застосунок має підтримувати систему управління базою користувачів для забезпечення ефективної взаємодії з клієнтами. Перед отриманням доступу до функціоналу сервісу користувач повинен пройти процедуру автентифікації, а у випадку відсутності облікового запису – зареєструватися. Результати виконаних аналізів мають зберігатися у внутрішній пам'яті системи, що дозволяє користувачеві переглядати історичну аналітику за попередніми запитами.

Для користування сервісом клієнт повинен заздалегідь підготувати й обробити текстові дані, які він бажає піддати емоційній класифікації.

Веб-застосунок повинен забезпечувати автоматичну агрегацію підсумкової статистики, що відображає емоційну забарвленість текстових даних. Отримана статистика має бути представлена у зручній та візуально зрозумілій формі, що сприятиме її ефективному застосуванню в бізнес-процесах компаній.

Під час реалізації функціональних вимог до програмного забезпечення необхідно врахувати низку обмежень, зокрема:

- підтримуються лише текстові дані англійською мовою;
- одна електронна пошта може бути прив'язана лише до одного облікового запису;
- додавання завдання можливе лише за наявності текстових даних у визначеному форматі;
- результат обробки доступний лише після завершення роботи нейронної мережі;
- один користувач може створювати необмежену кількість завдань.

До нефункціональних вимог, що висуваються до веб-застосунку, належать:

- архітектура системи повинна відповідати принципам модульності та слабого зв'язку між компонентами, що забезпечить легкість масштабування та інтеграції з зовнішніми сервісами, включаючи можливість підключення до баз даних для перенесення сгенерованих структур;

- система повинна мати достатню продуктивність для обробки великої кількості паралельних запитів, що вимагає підтримки механізмів кластеризації та масштабування як у вертикальному, так і в горизонтальному напрямках;

- конфіденційність та безпека даних повинні бути гарантовані, з обмеженням доступу лише для власника відповідної інформації.

Крім того, у процесі реалізації функціональності мають бути враховані такі технічні обмеження:

- навчання нейронної мережі повинно здійснюватися з високою швидкістю;

- серверна частина застосунку повинна бути здатною витримувати значні навантаження для забезпечення стабільної роботи системи;

- база даних не повинна мати обмежень щодо кількості записів, які може зберігати один користувач;

- унікальна ідентифікація клієнтів має здійснюватися на основі перевірки електронної адреси на унікальність.

Для здійснення класифікації текстових даних необхідно розробити нейронну мережу з використанням заздалегідь визначеної архітектури, реалізованої за допомогою фреймворку Keras. Модель має проходити навчання на попередньо розміченому наборі текстових даних.

З метою зберігання інформації про кожне окреме завдання доцільно використати систему керування базами даних. Оптимальним вибором у цьому випадку є СУБД SQLite, яка забезпечує ефективну інтеграцію з фреймворком Flask [12] та працює в поєднанні з ORM-бібліотекою SQLAlchemy.

Серверна частина відповідає за обробку текстових даних та їх збереження в базі. Архітектура застосунку повинна передбачати чітке розділення логічного рівня та рівня доступу до даних. Презентаційний рівень (шар представлення) слід виділити окремо.

Для реалізації візуальної частини інтерфейсу доцільно використати шаблонізатор Jinja2, що забезпечує сучасний і гнучкий механізм формування представлення у клієнт-серверних застосунках. Його застосування сприяє спрощенню та прискоренню процесу розробки, завдяки компактному синтаксису, який знижує кількість необхідних дій з боку розробника, водночас підвищуючи читабельність як HTML-розмітки, так і програмного коду.

Було проаналізовано ключові вимоги до розробки програмної системи для автоматизованої класифікації емоцій у текстових даних. Визначено, що критично важливим фактором ефективності такого сервісу є точність класифікації, яка досягається шляхом використання нейронної мережі з відповідною архітектурою, реалізованою за допомогою фреймворку Keras.

Для навчання моделі необхідне попереднє збирання, маркування та предобробка текстових даних, що забезпечує її здатність до узагальнення.

Система повинна підтримувати гнучку рольову модель користувача з обов'язковою автентифікацією та можливістю збереження результатів попередніх завдань. Важливою є реалізація зручного користувачького інтерфейсу, який поєднує сучасний дизайн із інтуїтивно зрозумілою навігацією, що підвищує якість взаємодії з кінцевим користувачем.

Функціональні та нефункціональні вимоги до веб-застосунку чітко окреслюють обмеження на рівні підтримки мови (лише англійська), унікальності облікового запису, вимоги до масштабованості, продуктивності та безпеки. Було запропоновано архітектурне рішення з чітким розділенням рівнів логіки, даних і представлення, реалізоване за допомогою Flask, SQLAlchemy, SQLite та шаблонізатора Jinja2, що дозволяє ефективно розвивати та масштабувати систему в майбутньому.

Загалом, запропоноване рішення дозволяє створити стабільний, безпечний і продуктивний веб-застосунок, придатний для впровадження в реальні бізнес-процеси з метою аналізу емоційної складової текстових відгуків і повідомлень.

3 АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 UML проектування ПЗ

Моделювання є методом дослідження об'єктів, процесів і явищ, який полягає в заміні реального об'єкта його аналогом – моделлю, що зберігає основні характеристики та властивості оригіналу. Під моделюванням розуміють процес побудови й аналізу моделей, які виступають інструментом та формою наукового пізнання. Існує низка підходів до моделювання, однак у контексті створення програмного забезпечення перевага надається математичному моделюванню. Такий тип моделювання базується на побудові моделей у вигляді систем співвідношень, що відображають функціонування певних технологічних, економічних чи інших процесів.

Уніфікована мова моделювання UML (Unified Modeling Language) є загальноприйнятим стандартом у сфері об'єктно-орієнтованого програмування та становить невід'ємну складову уніфікованого процесу розробки програмного забезпечення [13]. UML дозволяє створювати абстрактні графічні моделі систем, які називають UML-моделями, і використовується для формалізації, візуалізації, проектування та документування програмних систем.

UML може бути застосована на всіх етапах життєвого циклу розробки програмного забезпечення – від аналізу бізнес-вимог до проектування прикладних рішень. Широкий спектр діаграм, підтримуваних цією мовою, дозволяє зобразити архітектуру системи у вигляді, придатному до трансляції в програмний код об'єктно-орієнтованих мов, що значно прискорює процес розробки. Саме тому для формалізації структури та логіки роботи веб-застосунку було побудовано відповідні UML-діаграми, які наочно і повно відображають архітектуру програмної системи.

Проектування UML-діаграм здійснювалося за допомогою інструменту Lucidchart, який забезпечує зручне середовище для створення візуальних моделей систем [14].

Діаграма варіантів використання. Для розробки системи необхідно провести проектування поставлених задач, що включає в себе моделювання системи та її компонентів за допомогою UML-діаграм. На рисунку 3.1 представлена діаграма варіантів використання, на якій зображено актора, що є зареєстрованим користувачем системи. Гість має можливість зареєструватися або увійти до системи.



Рисунок 3.1 – Мапа користувача

Авторизований користувач має змогу створювати нові завдання для веб-застосунку, які полягають у проведенні аналізу текстових даних, або переглядати вже виконані завдання. При створенні нового завдання

користувач завантажує файл і чекає, поки веб-застосунок відобразить результати.

На етапі початкового аналізу була визначена основна структура та функціонування системи, а також взаємодія з нею з точки зору різних категорій користувачів, зокрема зареєстрованих та незареєстрованих користувачів.

3.2 Проектування архітектури ПЗ

Наступним етапом розробки веб-застосунку є проектування його архітектури. Проектування повинно здійснюватися з використанням трирівневої архітектури, що ґрунтується на принципі «Закону Деметри» (Law of Demeter, LoD) [15], основна ідея якого полягає в тому, що об'єкт має мати мінімальне знання про структуру та властивості інших об'єктів, зокрема своїх підкомпонентів.

Класична трирівнева архітектура складається з таких рівнів: рівень представлення, рівень бізнес-логіки та рівень доступу до даних.

Рівень представлення (Presentation layer) є тим рівнем, з яким безпосередньо взаємодіє користувач. Цей рівень включає компоненти для створення інтерфейсу користувача та механізм отримання введення від користувача.

Рівень бізнес-логіки (Business layer) містить компоненти, які відповідають за обробку даних, отриманих з рівня представлення. Він реалізує основну логіку додатка, здійснює необхідні обчислення, взаємодіє з базою даних і передає результат обробки на рівень представлення.

Рівень доступу до даних (Data Access layer) відповідає за збереження моделей, що описують сутності, а також містить специфічні класи для роботи з різними технологіями доступу до даних. Цей рівень також включає репозиторії, через які рівень бізнес-логіки взаємодіє з базою даних.

Згідно з принципами архітектури, крайні рівні не можуть безпосередньо взаємодіяти між собою, тобто рівень представлення не може звертатися безпосередньо до бази даних або навіть до рівня доступу до даних. Взаємодія між ними здійснюється виключно через рівень бізнес-логіки.

Для забезпечення якісної роботи веб-застосунку необхідно також виконати проектування архітектури нейронної мережі.

Архітектура нейронної мережі складається з кількох взаємопов'язаних шарів:

- embedding;
- bidirectional long short-term memory (LSTM);
- pooling;
- dropout;
- dense layers.

Шар embedding є основною формою представлення слів, що зв'язує людське розуміння мови з інтерпретацією машини. Цей тип шару створює розподілені представлення тексту в n-вимірному просторі, що є критично важливим для вирішення більшості завдань обробки природної мови.

Word embedding являє собою набір методів обробки природної мови, спрямованих на відображення семантичного значення слів у геометричному просторі. Це досягається шляхом асоціації кожного слова в словнику з числовим вектором таким чином, щоб відстань (наприклад, L2 відстань або більш часто – косинусна відстань) між векторами двох слів відображала частину їхніх семантичних відносин. Геометричний простір, що формується цими векторами, називається простором вкладення.

Наприклад, слова «кокос» і «білий ведмідь» будуть представлені як вектори, що знаходяться на великій відстані один від одного через семантичну різницю, тоді як слова «кухня» та «вечеря», які мають близьке значення, будуть розміщені поруч.

Методи для обчислення word embeddings включають зменшення розмірності наборів даних, що містять статистику спільного входження слів у текстовому корпусі, і застосовуються через нейронні мережі, такі як «word2vec» або методи матричної факторизації.

LSTM-мережа (Long Short-Term Memory) – це тип нейронної мережі, що використовує LSTM-модулі замість або разом з іншими мережевими компонентами. LSTM-модулі є рекурентними компонентами мережі, здатними зберігати інформацію на різних часових інтервалах.

Ключова особливість LSTM-модулів полягає в тому, що вони не використовують стандартну функцію активації всередині своїх рекурентних компонентів, що дозволяє зберігати значення без втрат на протязі часу. Це дозволяє уникнути розмивання збережених значень і зберегти ефективність навчання через зворотне поширення помилок у часі (Backpropagation through time) [16].

Двонаправлена рекурентна нейронна мережа (BRNN) об'єднує два прихованих шари, що з'єднані в протилежних напрямках з однаковим входом, як показано на рисунку 3.2. Завдяки такій структурі мережа може одночасно отримувати інформацію як з попередніх, так і з наступних станів.

Це дозволяє мережі враховувати контекст з обох напрямків, що значно покращує її здатність до навчання. BRNN були розроблені для розширення обсягу вхідної інформації, доступної для мережі. Наприклад, багатошаровий перцептрон (multilayer perceptron) та нейронна мережа з часовою затримкою (time delay neural network) мають обмеження на гнучкість вхідних даних, оскільки ці мережі вимагають фіксації вхідних даних [16], [17].

Стандартна рекурентна нейронна мережа (RNN) має свої обмеження, оскільки майбутня вхідна інформація не може бути доступною з поточного стану. В протилежність цьому, BRNN не вимагає фіксації вхідних даних і надає можливість доступу до майбутньої інформації з поточного стану.

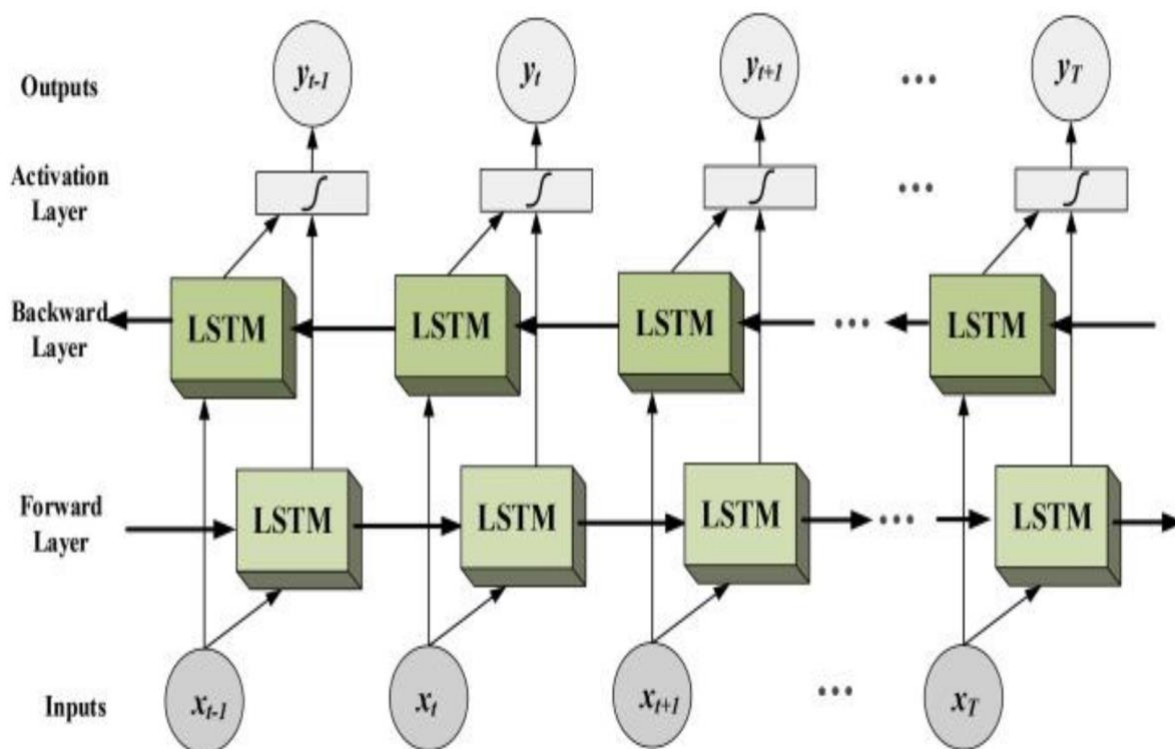


Рисунок 3.2 – Схема BRNN

BRNN є особливо корисною, коли важливий контекст вхідних даних для покращення результату. Наприклад, при розпізнаванні рукописного тексту точність може бути покращена завдяки врахуванню літер, що розташовані до і після поточної [16].

Шар пулінга, або підвибірки, є нелінійним методом ущільнення карти ознак, при якому група елементів (зазвичай розміру 2×2) об'єднується в один елемент шляхом нелінійного перетворення. Найпоширенішою функцією є функція максимуму, яка вибирає елемент з найбільшим значенням у кожному підмасиві. Операція пулінга дозволяє значно зменшити розмір даних і зберегти лише найбільш важливі ознаки, що запобігає перенавчанню моделі. Крім того, можна використовувати інші функції, такі як середнє значення або L2-нормування [18].

Після операції пулінга дані передаються в звичайну повнозв'язну нейронну мережу, де втрачається просторову структуру, і зменшується розмірність. Для покращення здатності до навчання перед вихідним шаром

було введено шар дропаута. Дропаута – це метод регуляризації, який запобігає перенавчанню, шляхом випадкового виключення деяких нейронів під час навчання, що значно підвищує швидкість навчання [19].

Мережі для навчання використовують метод виключення нейронів з ймовірністю p , таким чином, ймовірність того, що нейрон залишиться в мережі, становить $q = 1-p$. «Виключення» нейрона означає, що при будь-яких вхідних даних або параметрах він повертає 0. Виключені нейрони не беруть участь у процесі навчання на жодному з етапів алгоритму зворотного поширення помилки, тому виключення хоча б одного нейрона еквівалентне навчанню нової нейронної мережі.

Вихідним шаром було вирішено зробити повнозв'язний шар, кількість нейронів в якому дорівнює кількості емоцій, що класифікуються. Для функції активації нейронів було обрано softmax, яка є узагальненням логістичної функції для багатовимірного випадку. Ця функція перетворює вектор z розмірності K у вектор σ тієї ж розмірності, де кожен елемент σ_i вектору представлений дійсним числом в інтервалі $[0, 1]$, і сума всіх елементів дорівнює 1. Формула для функції Softmax має вигляд:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{k=1}^N e^{z_k}}, \quad (3.1)$$

де z_i – значення на виході з i -го нейрона до активації;

N – загальна кількість нейронів в шарі.

Функція Softmax застосовується в машинному навчанні для задач класифікації, коли кількість можливих класів більше двох (для двокласових задач використовують логістичну функцію). Координати i отриманого вектора інтерпретуються як ймовірності того, що об'єкт вже належить до класу i .

Таким чином, була спроектована нейронна мережа, архітектуру якої можна побачити на рисунку 3.3. В якості функції втрат для цієї моделі було

вибрано категоріальну крос-ентропію. Категоріальна крос-ентропія є функцією втрат, яка використовується для категоризації з однією міткою: коли кожній точці даних застосовується лише одна категорія. Тобто, приклад може належати тільки одному класу.

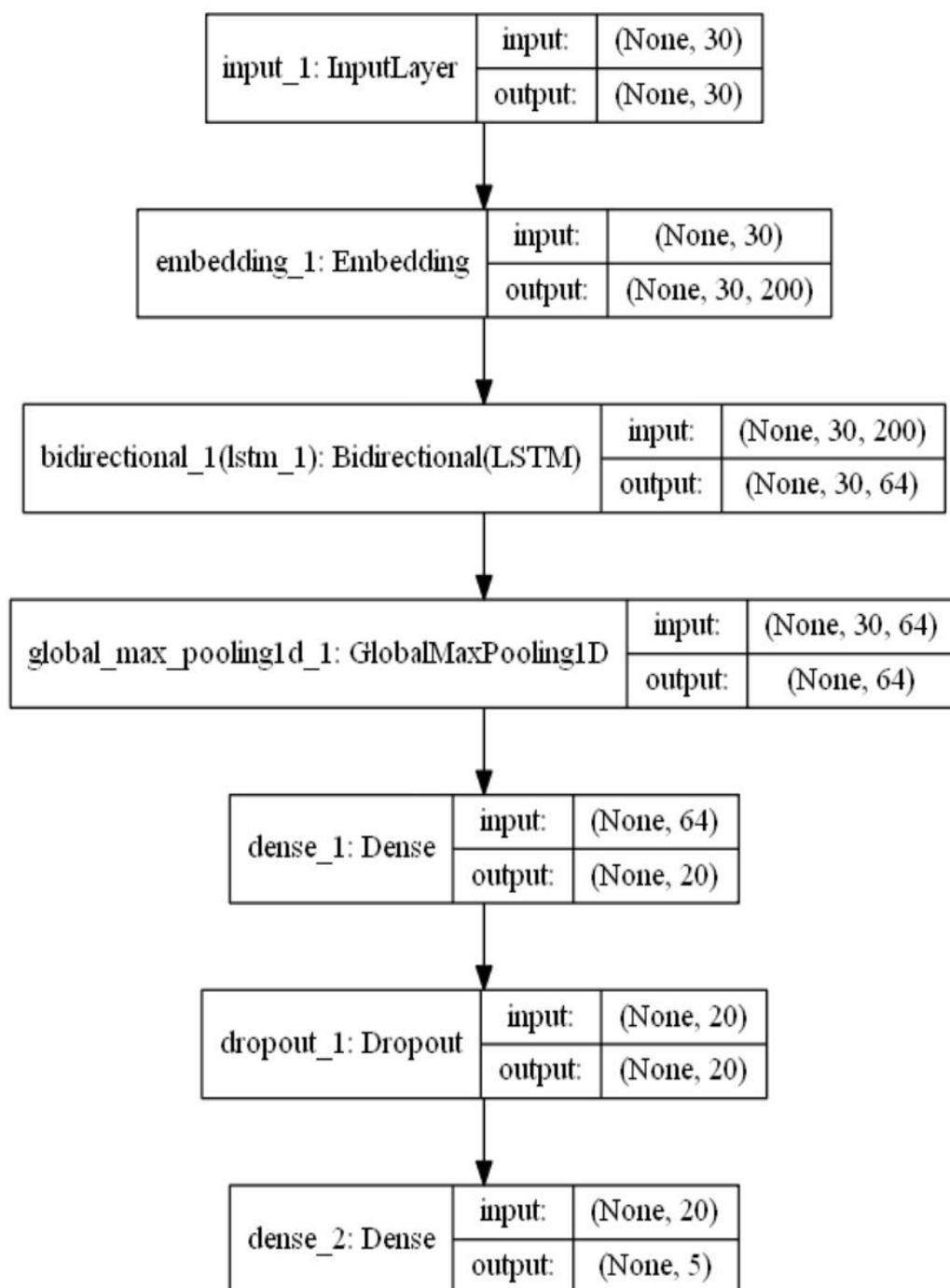


Рисунок 3.3 – Архітектура нейронної мережі

Категоріальна крос-ентропія порівнює розподіл пророкувань (активацій на вихідному шарі, по одному для кожного класу) з істинним розподілом, де ймовірність істинного класу встановлена рівною 1, а для інших класів – рівною 0. Іншими словами, істинний клас представлений у вигляді one-hot закодованого вектора, і чим ближче вихідні дані моделі до цього вектора, тим менші втрати.

Категоріальна крос-ентропія використовується разом із функцією активації Softmax, оскільки остання перетворює вихідні значення моделі на ймовірнісні оцінки, що є необхідною умовою для коректного обчислення втрат у задачах багатокласової класифікації.

3.3 Проектування структури зберігання даних

Під час проектування архітектури програмної системи для підвищення безпеки було ухвалено рішення розмістити базу даних безпосередньо поруч із бізнес-шаром. У цій базі даних повинна зберігатися інформація про користувачів і результати проведених аналізів. Таке розділення компонентів системи дозволяє підвищити як її надійність, так і безпеку доступу до даних [20].

Структура основної бази даних відображена на рисунку 3.4 і одночасно виконує функцію діаграми класів, оскільки моделі, які взаємодіють з базою даних, є об'єктами відповідних класів. Таким чином, робота з класами напряму пов'язана з обробкою даних у базі.

Таблиця user призначена для зберігання персональної інформації користувачів, необхідної для проходження авторизації та подальшої взаємодії з системою. До таких даних належать електронна пошта, ім'я та пароль користувача.

У таблиці summary фіксується інформація про всі виконані аналізи текстових даних для кожного користувача. Кожен запис містить ідентифікатор користувача, опис завдання, дату та час запиту, а також

узагальнену статистику щодо виявлених емоцій – зокрема, кількість текстів, у яких переважає певна емоція.

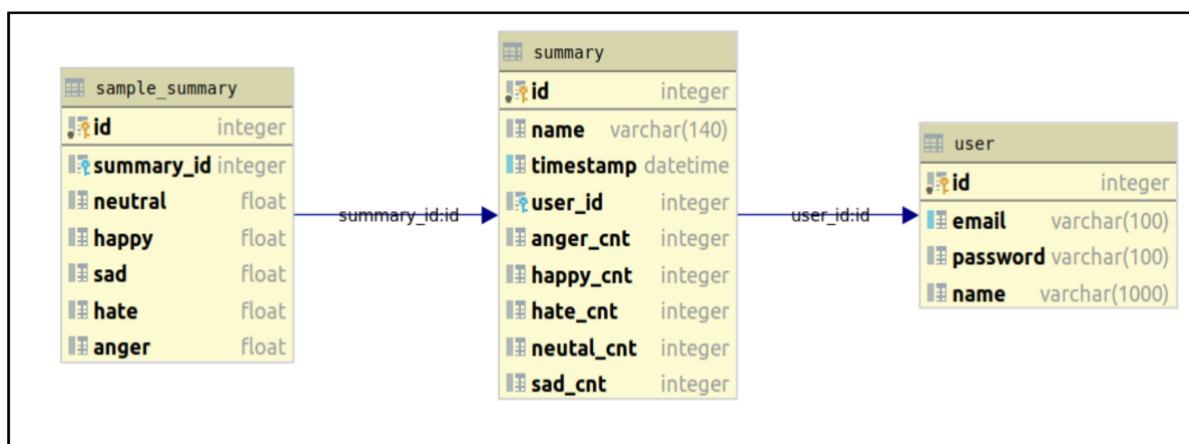


Рисунок 3.4 – Структура бази даних

Таблиця `sample_summary` зберігає детальну інформацію про результати аналізу емоційного наповнення кожного окремого текстового примірника. Вона містить посилання на відповідний запис у таблиці `summary`, що вказує на загальний аналіз, у межах якого було оброблено даний текст. Крім того, таблиця містить п'ять полів, кожне з яких відповідає певній емоції.

Кожне з цих полів містить числову оцінку, що відображає рівень присутності відповідної емоції в тексті. Значення подано у форматі типу `float` і варіюється в межах від 0 до 1, де 0 означає повну відсутність емоції, а 1 – її максимальну присутність.

3.4 Аналіз існуючих алгоритмів та методів

У якості шару ембедінга в моделі було використано попередньо навчені векторні представлення слів GloVe (Global Vectors) [21]. Це алгоритм навчання без учителя, який будує векторні уявлення слів на основі агрегованої глобальної статистики співвходження слів у корпусі. На відміну

від моделей, що працюють лише з локальними контекстами (наприклад, word2vec), GloVe використовує повну матрицю співвідношення, що дає змогу моделі виявляти більш глибокі семантичні зв'язки між словами.

Навчання GloVe здійснюється на ненульових елементах цієї матриці, що значно пришвидшує обчислення після початкового збору статистики. Для створення таких векторів потрібен лише один прохід по корпусу. У рамках даної моделі використовувалась уже готова, попередньо заповнена матриця з векторами, що зменшило обчислювальні витрати та скоротило час підготовки.

Навчання нейронної мережі здійснюється з використанням алгоритму зворотного поширення помилки. Цей алгоритм включає два етапи: прямий та зворотний прохід. На етапі прямого проходу вхідні дані передаються через всі шари мережі, формуючи вихідний сигнал. На етапі зворотного проходу обчислюється різниця між фактичним виходом мережі та очікуваним результатом – сигнал помилки. Цей сигнал поширюється у зворотному напрямку по мережі, і на основі нього коригуються синаптичні ваги, щоб зменшити помилку на наступних ітераціях. Таким чином, модель поступово навчається наближати свій вихід до правильного результату.

Алгоритм зворотного поширення помилки включає наступні кроки:

- ініціалізувати синаптичні ваги невеликими випадковими значеннями;
- вибрати чергову навчальну пару з навчальної вибірки та подати вхідний вектор на вхід мережі;
- обчислити вихід мережі;
- визначити різницю між фактичним виходом мережі та цільовим вектором (необхідним виходом);
- скоригувати ваги мережі з метою мінімізації помилки;
- повторювати кроки 2–5 для кожного вектора навчальної множини до досягнення прийнятного рівня помилки на всьому наборі даних.

Для оцінки точності роботи нейромережі під час навчання використовувалась метрика accuracy – частка правильно класифікованих прикладів від загальної кількості.

3.5 UI/UX Design

UX (User Experience – досвід користувача) – це загальне враження, яке отримує користувач під час взаємодії з інтерфейсом. Ця область відповідає за те, наскільки ефективно, швидко і просто користувач може досягти своєї мети при використанні продукту.

UI (User Interface – інтерфейс користувача) – це візуальна складова інтерфейсу, тобто те, як він виглядає і як взаємодіє з користувачем. UI визначає колірну гаму, типографіку, розміри кнопок, відступи, зручність натискання елементів, читабельність тексту тощо.

UX/UI дизайн – це проектування користувацьких інтерфейсів, у якому однаково важливими є як зручність використання, так і естетичне сприйняття.

На початковому етапі проектування було проведено аналіз сучасних тенденцій у сфері веб-дизайну. Було вивчено популярні рішення, новаторські дизайнерські підходи, а також проведено огляд провідних вебсайтів за 2019 рік. Це дозволило визначити основні принципи створення інтерфейсу, які поєднують зручність користування та привабливий зовнішній вигляд.

Тому під час проектування дизайну було вирішено дотримуватися таких принципів:

- інтерфейс повинен бути впізнаваним, а його призначення – очевидним для користувача;
- користувачі повинні чітко розуміти, з чим саме вони взаємодіють;
- поведінка інтерфейсу має бути передбачуваною;
- світло візуально має падати згори (імітуючи природне освітлення);

- спочатку проєктуються елементи в чорно-білій гамі;
- максимальне використання білого простору;
- накладання тексту на зображення;
- застосування сучасних популярних шрифтів.

Білий робочий простір – загальноприйнята дизайнерська техніка, яка знижує візуальну перевантаженість, зменшує навантаження на очі та створює ефект простору. Це особливо важливо для користувачів мобільних пристроїв або невеликих екранів.

Принцип «спочатку чорне і біле» передбачає, що під час первинного проєктування інтерфейсу потрібно обмежитися лише чорно-білою кольоровою гамою. Це дозволяє зосередитися на структурі, логіці взаємодії та читабельності. Лише після цього додається один яскравий акцентний колір (наприклад, помаранчевий, червоний або зелений), який слугує для виділення ключових елементів – кнопок, закликів до дії, маркетингових блоків тощо.

Збільшений білий простір. Щоб інтерфейс виглядав привабливо та «дихав», необхідно використовувати достатню кількість білого простору. Це – загальновизнаний прийом у сучасному UX/UI-дизайні. Білий простір зменшує візуальну напругу під час тривалої взаємодії з інтерфейсом, підвищує комфорт користувача та створює ілюзію відкритості та простору. Особливо це актуально для мобільних пристроїв з невеликими екранами.

Виділення тексту як інструмент фокусування уваги. Контраст – ключ до ефективного візуального сприйняття тексту. Один з прийомів – збільшити розмір шрифту, зберігаючи при цьому його легкість і витонченість. Такий баланс дозволяє одночасно привертати увагу та зберігати естетичну гармонію інтерфейсу.

Використання сучасних шрифтів. У випадку загального дизайну сайтів доцільно використовувати універсальні, сучасні шрифти, що легко читаються. Найпопулярнішими останніми роками стали Roboto, Open Sans і

Helvetica Neue. Вони мають просту структуру, добре адаптуються до різних екранів та не відволікають увагу від основного контенту.

Згідно з новітніми дизайнерськими трендами, зображення все частіше замінюють великими акцентними написами. Такий підхід дозволяє економити простір, спрощує адаптацію інтерфейсу та надає йому виразного стилю.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ

Оскільки проект є веб-застосунком, він включає як серверну, так і клієнтську частину. Серверна частина відповідає за обробку введених користувачем даних та надання відповідної інформації на основі запитів клієнта. Уся інформація зберігається у базі даних на сервері, а сервер передає її клієнту при необхідності. Веб-клієнт, у свою чергу, забезпечує динамічну навігацію між сторінками та інтерфейсну взаємодію з користувачем.

Для реалізації серверної частини було обрано технологію створення веб-сервісів Flask. Робота з базою даних, створеною на основі Sqlite, здійснюється з використанням бібліотеки SQLAlchemy. Основною мовою програмування було обрано Python 3.6 через її популярність та широкий вибір бібліотек для веб-розробки. У якості середовища розробки було використано PyCharm, оскільки воно забезпечує зручний доступ до інструментів та інтеграцію з Python-бібліотеками.

Для клієнтської частини використовувались такі мови та технології: HTML і CSS – для структурування та стилізації сторінок; Jinja2 – шаблонізатор для Python, що дозволяє легко поєднувати HTML-код із серверною логікою; Bulma – сучасна CSS-бібліотека, що надає зручні інструменти для створення адаптивного (responsive) дизайну, а також включає набір іконок та елементів інтерфейсу.

Оскільки шаблони Jinja працюють на основі текстового формату, їх створення схоже на написання звичайного коду. Вони дозволяють гнучко налаштовувати теги, фільтри, тести та глобальні параметри, що значно спрощує генерацію HTML-розмітки та її адаптацію до даних, отриманих із сервера.

Повертаючись до взаємодії сервера та клієнта, логіка роботи веб-застосунку була організована таким чином, що існує шар контролера, який відповідає за взаємодію між шаром даних і шаром відображення, а також

обробку клієнтських запитів до сторінок. Іншими словами, коли користувач звертається до певної сторінки, спрацьовує відповідний метод контролера. Цей метод отримує базову інформацію про сесію, наприклад, чи авторизований користувач, а також збирає дані, необхідні для формування вмісту сторінки. Далі контролер передає зібрану інформацію шару подання, сигналізуючи, що дані готові до відображення.

Шар подання, отримавши дані від контролера, формує HTML-шаблон сторінки, а також підключає необхідні файли, які потрібні для роботи клієнтської частини. Потім до роботи підключається клієнтський код. Він обробляє інформацію, яка була передана сервером під час генерації сторінки, запускає клієнтську логіку та, у разі потреби, асинхронно завантажує додаткові ресурси або використовує вже збережені у кеші. Після завершення всіх етапів сторінка стає готовою до взаємодії з користувачем.

Прикладом такого підходу є метод `login` контролера `auth`, який відповідає за відображення сторінки авторизації. Він збирає усі необхідні дані та повертає готову до показу веб-сторінку. Коли користувач вводить свої дані та підтверджує авторизацію, спрацьовує метод `login_post`. Цей метод обробляє введену інформацію, перевіряє її на відповідність збереженим у базі даних обліковим записам і, в разі успішного збігу, авторизує користувача або ж повідомляє про помилку, якщо дані некоректні або такий користувач не знайдений.

Більша частина роботи щодо вилучення інформації з бази даних виконується в моделі, яка реалізує шар даних. Для кожної таблиці бази даних створюється окрема модель, яка відповідає за взаємодію саме з цією таблицею. Логіка роботи таких моделей, як правило, проста, особливо коли потрібно лише отримати дані з бази в «чистому» вигляді. Наприклад, модель `User` відповідає за таблицю `user` і дозволяє працювати з її даними.

Логіка клієнтської частини реалізована з використанням технології шаблонів `Jinja2`. У клієнтському коді реалізується реакція на дії користувача, тому `Jinja2` дозволяє частину обчислень виконувати на стороні

клієнта, а також робить веб-клієнт інтерактивним. Завдяки можливості інтеграції коду безпосередньо в HTML, Jinja2 значно спрощує динамічне формування сторінок.

Для створення сучасного, гнучкого та адаптивного інтерфейсу було використано CSS-бібліотеку Bulma. Вона дозволяє забезпечити коректне відображення сайту на пристроях з різними розмірами екранів. Залежно від розміру дисплею змінюються розміри і позиції елементів, що дозволяє максимально ефективно використовувати доступну площу. Крім того, з бібліотеки Bulma були використані готові елементи інтерфейсу, на основі яких побудовано дизайн сайту.

Таким чином, клієнтська частина веб-застосунку реалізована так, щоб бути максимально зручною, адаптивною та інтерактивною, відповідаючи сучасним вимогам користувача. Серверна частина відповідає за взаємодію з базою даних, обробку вхідних і вихідних даних, перетворення їх у потрібний формат, а також за передачу інформації між клієнтом і сервером.

Окремо був створений модуль `nn_tools`, який дозволяє здійснювати взаємодію з нейронною мережею. Цей модуль був розроблений на основі бібліотеки Keras і надає контролерам інтерфейс для обробки даних за допомогою моделі.

Для навчання нейронної мережі був обраний датасет з відкритого джерела на GitHub. Він містить 47288 прикладів розмічених твітів, які класифіковані за п'ятьма емоційними станами: радість, смуток, злість, агресія та нейтральна емоція. Кількість прикладів у кожному класі становить: `neutral` – 9643, `happy` – 16297, `sad` – 15938, `hate` – 4301, `anger` – 1109. Датасет було вибрано завдяки його великому обсягу і наявності чітких міток для кожного запису.

Усі тексти в наборі були попередньо оброблені. Було проведено нормалізацію – очищення від розділових знаків, переведення в нижній регістр, лематизацію – приведення слів до їхньої базової форми, а також видалення стоп-слів.

Наступним етапом підготовки даних до навчання стало приведення їх до форми, яка є зручною для подачі в якості входних параметрів нейронної мережі. Для цього було проведено векторизацію тексту за допомогою засобів бібліотеки Keras. Кожному слову в тексті був зіставлений числовий індекс у словнику. У результаті кожен зразок вибірки перетворювався на вектор чисел. Щоб довжина тексту не впливала на здатність нейронної мережі до узагальнення, кожен вектор було доповнено нулями до сталої довжини.

Векторизоване уявлення даних було розділено на дві множини – тестову та тренувальну – у співвідношенні 1 до 3. Тренувальна вибірка використовувалася для навчання моделі, а тестова – для перевірки її якості на нових, не бачених під час навчання даних.

Для реалізації моделі було використано бібліотеку Keras. Як архітектура була обрана спеціально спроектована нейронна мережа. На вхід подавалися вектори токенів сталої довжини, які передавалися через Embedding-шар. Вихід цього шару мав розмірність 30 на 200, де кожне слово відображалось у вигляді вектора з 200 компонентів, а 30 – це фіксована довжина послідовності.

Після цього дані проходили через шар двонаправленої нейронної мережі, яка складається з двох незалежних RNN. Для однієї з них входна послідовність подавалась у прямому порядку, а для іншої – у зворотному. Виходи обох мереж об'єднувались на кожному часовому кроці, що дозволяло моделі мати як прямий, так і зворотний контекст на всіх етапах обробки. Результатом цього шару була матриця розміром 30 на 64.

Далі застосовувався шар GlobalMaxPooling1D, який дозволяв зменшити розмірність даних, взявши максимум по кожному векторному виміру. Це суттєво зменшувало обсяг пам'яті, яку використовує модель, і пришвидшувало процес навчання, позбавляючи модель від надмірної інформації.

На завершальному етапі вихідні дані проходили через два повнозв'язні (Dense) шари, між якими застосовувалося відключення (dropout) 5% нейронів. Така операція дозволяла запобігти перенавчанню моделі, роблячи її більш стійкою до нових даних.

Останній шар нейронної мережі містить 5 нейронів, що відповідає кількості класів емоцій. Для кожного з них використовується функція активації softmax, яка дозволяє перетворити вихідні значення в імовірності належності до певного класу. Це забезпечує зручну інтерпретацію результатів класифікації.

На етапі тестування нейронної мережі була проведена оцінка якості отриманої моделі за допомогою тестового набору даних, що складався з 9457 текстових записів з мікроблогінгового сервісу Twitter. Як метрику обрано F-міру, яка є гармонійним середнім між precision (точністю) та recall (повнотою). Precision вимірює частку правильно класифікованих випадків серед усіх передбачених як позитивні, а recall – частку правильно класифікованих випадків серед усіх релевантних.

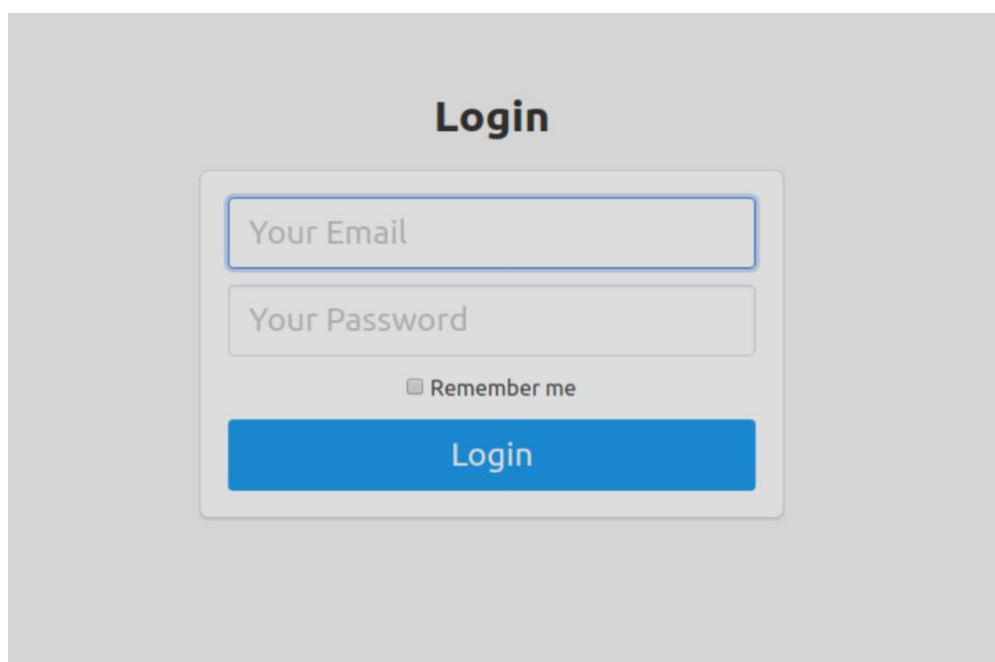
Отримане значення F-міри на тестовому наборі склало 0.62, що є гідним результатом з урахуванням складності завдання та порівняно з існуючими промисловими рішеннями у подібних задачах.

Інтерфейс користувача починається з початкової сторінки, на якій представлено можливість входу в систему. Після натискання кнопки авторизації користувач переходить на сторінку входу рисунок 4.1.

Ця сторінка передбачає введення електронної пошти та пароллю для авторизації. Після введення даних користувач натискає кнопку входу, після чого, у разі правильного введення, переходить на свою персональну початкову сторінку. Якщо дані введено некоректно, система виводить повідомлення про помилку авторизації.

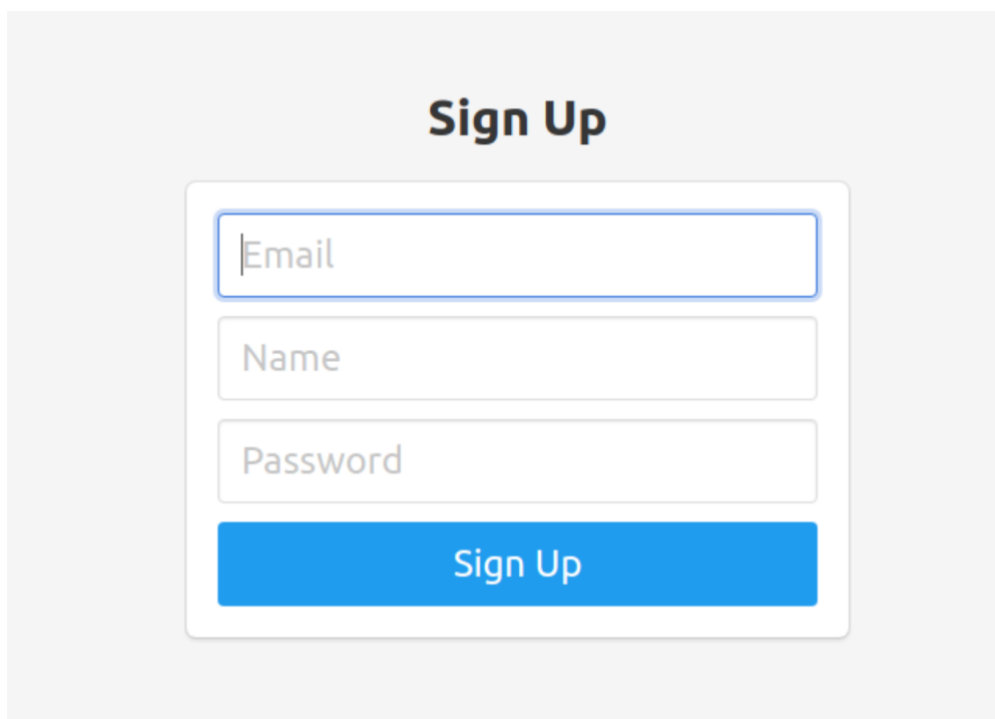
Для того щоб мати змогу увійти на сайт, користувач повинен попередньо зареєструватися. У процесі реєстрації необхідно ввести

особисті дані, вказати електронну пошту та створити пароль. На рисунку 4.2 зображено відповідну форму реєстрації.



The image shows a login form titled "Login" centered on a light gray background. The form is contained within a white rounded rectangle. It features two input fields: "Your Email" and "Your Password". Below the password field is a checkbox labeled "Remember me". At the bottom of the form is a blue button with the text "Login" in white.

Рисунок 4.1 – Авторизація користувача



The image shows a sign-up form titled "Sign Up" centered on a light gray background. The form is contained within a white rounded rectangle. It features three input fields: "Email", "Name", and "Password". At the bottom of the form is a blue button with the text "Sign Up" in white.

Рисунок 4.2 – Реєстрація користувача

Якщо дані було введено коректно, то інформація про користувача додається до бази даних. Зареєстровані користувачі можуть входити до веб-застосунку за допомогою форми авторизації.

Після авторизації користувач потрапляє на головну сторінку веб-застосунку, де відображаються всі попередні аналізи рисунок 4.3. Вони представлені у вигляді таблиці, яка містить імена файлів з текстами та дати відповідних запитів. Якщо натиснути на стрічку таблиці, відкриється сторінка з детальною аналітикою цього конкретного файлу.

Крім того, користувач має можливість виконати новий запит прямо з головної сторінки. Для цього потрібно обрати файл із підготовленими даними та натиснути кнопку «upload». Після завантаження файлу відкриється сторінка з докладним описом емоційної складової текстових даних, що містилися у файлі.

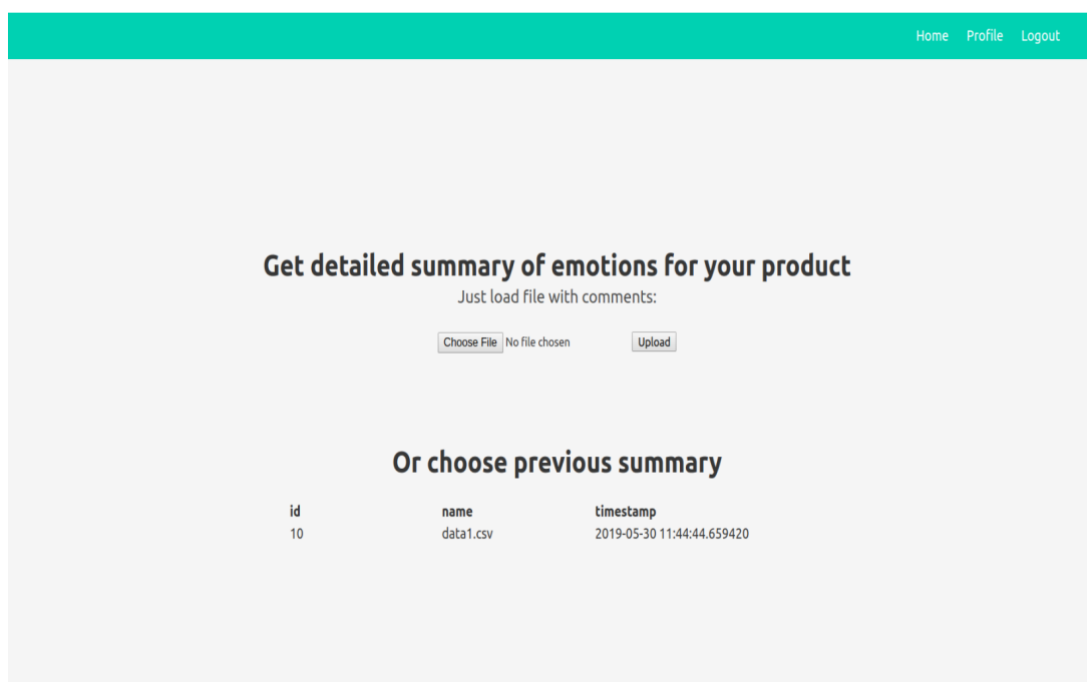


Рисунок 4.3 – Головна сторінка

Сторінка з аналітикою, яка зображена на рисунку 4.4, містить інформацію про кількість текстових екземплярів, у яких найбільше

виражена одна з емоцій: радість, смуток, агресія, злість або нейтральна емоція. Кожна з емоцій позначена відповідним символічним зображенням для зручності сприйняття.

Якщо користувач бажає переглянути детальні дані щодо кожного окремого текстового екземпляру з файлу, він може зробити це у таблиці, розташованій під загальною статистикою. У цій таблиці кожен рядок представляє окремий текст і містить п'ять числових значень – по одному для кожної емоції. Ці значення відображають ймовірність присутності відповідної емоції в тексті. Значення варіюються від 0 до 1, де 0 означає повну відсутність емоції, а 1 – що саме ця емоція є домінуючою в даному тексті.



Рисунок 4.4 – Аналітика

Користувач має можливість вийти з веб-застосунку, натиснувши кнопку Logout. У результаті цього його сесія завершується, і для повторного входу до застосунку необхідно знову ввести електронну пошту та пароль.

5 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Тестування є дуже важливим етапом створення проекту, оскільки дозволяє вчасно виявити та усунути помилки реалізації, які не повинні потрапити до кінцевого користувача [22]. Під час розробки веб-застосунку було застосовано кілька видів тестування.

Інсталяційне тестування (Installation testing) передбачає перевірку коректності установки і деінсталяції програмного продукту в середовищі, максимально наближеному до експлуатаційного. Однак його не проводили, оскільки створюваний застосунок є веб-сайтом і не вимагає інсталяції на пристроях користувачів.

Димове тестування (Smoke testing) – це перший запуск програми після написання або внесення суттєвих змін. Його виконували кілька разів при додаванні нових модулів у програму.

Функціональне тестування (Functional testing) полягає у перевірці відповідності продукту функціональним вимогам і специфікаціям. Його проводили вручну після завершення розробки програмного продукту.

Регресійне тестування і ретестування (Regression and retesting) – це повторне тестування після внесення змін до програмного забезпечення або його оточення у новій версії, щоб переконатися, що раніше працюючі функції залишаються працездатними. Воно проводилося вручну. В ході регресійного тестування виявили декілька помилок, яким було надано пріоритет залежно від їх впливу на основний функціонал програми. Всі критичні та високопріоритетні помилки були виправлені.

Інтеграційне тестування (Integration testing) – це перевірка суміщених компонентів прикладної програми з метою визначення коректності їх спільної роботи. У проекті його було проведено програмно в межах тестових проектів.

Тестування графічного інтерфейсу користувача (User Interface testing) включає перевірку роботи інтерфейсу – сторінок, кнопок та інших

елементів. Його виконували після завершення розробки програмного забезпечення. Тестування показало, що веб-клієнт працює стабільно, адекватно реагує на зміну розширення екрану, а інтерфейс є інтуїтивно зрозумілим.

Тестування продуктивності (Performance testing) передбачає оцінку швидкодії системи – час відгуку, частоту транзакцій та інші параметри в імітаційних та реальних умовах. Веб-застосунок тестували на ПК, де проводилась розробка, і отримали прийнятні показники – відгук системи триває близько 1 секунди.

Стрес-тестування (Stress testing) є видом тестування продуктивності, при якому перевіряють, як система реагує на стресові ситуації, наприклад, нестачу ресурсів (дискового простору, обрив мережі тощо).

Тестування безпеки (Security testing) використовується для оцінки захищеності системи, виявлення ризиків, пов'язаних з несанкціонованим доступом, атаками хакерів, вірусами тощо. Було перевірено, чи має неавторизований користувач доступ до основних функцій веб-застосунку.

Тестування локалізації (Localization testing) – це перевірка правильності локалізації продукту: коректності перекладу інтерфейсних елементів, системних повідомлень, розділів «Допомога»/«Довідка» та документації. Оскільки веб-застосунок орієнтований на англомовних користувачів, тестування локалізації зосереджувалося на відповідності інтерфейсу нормам англійської мови.

Юзабіліті тестування (usability testing) – це метод, спрямований на оцінку зручності використання продукту, його навченості, зрозумілості та привабливості для користувачів у заданому контексті. Мета цього тестування – виявити проблеми, пов'язані зі специфікою інтерфейсу, оцінити, наскільки комфортно користувачам орієнтуватися в системі та використовувати основний функціонал.

Тестування стабільності та надійності (stability testing) – це вид автоматизованого тестування, який перевіряє працездатність додатку при

тривалому навантаженні. Воно дозволяє оцінити, наскільки система стійка під час довготривалої роботи.

У рамках проекту було протестовано всі класи бізнес-логіки, контролери в шарі відображення даних, взаємодію з базою даних, а також взаємодію з нейронною мережею. Рівень доступу до даних перевірявся за допомогою інтеграційних тестів.

ВИСНОВКИ

Під час написання кваліфікаційної роботи було створено веб-застосунок для класифікації тексту за допомогою нейромережевого підходу, що спрямований на розпізнавання емоцій у текстах.

Проведений аналіз предметної галузі допоміг обрати оптимальні технології для реалізації проекту: Flask, SQLAlchemy, SQLite, Keras, Jinja2, Vulma, мова програмування Python та середовище розробки PyCharm.

Було розроблено функціональні та нефункціональні вимоги до веб-застосунку, а також створено діаграму варіантів використання, яка демонструє взаємодію користувача із системою. Основна база даних була спроектована для зберігання інформації про користувачів.

В ході роботи спроектовано архітектуру нейронної мережі, проведено її навчання та оцінку якості класифікації текстів. Модель реалізована на Python із використанням бібліотеки Keras. Інтерфейс користувача розроблено таким чином, щоб не вимагати додаткового навчання для роботи з системою. Веб-застосунок зберігає базу користувачів і результати аналізу, що дозволяє переглядати попередні запити та їх аналітику.

Автоматично формується сумарна статистика емоційної забарвленості текстів, яка є зручною для використання в бізнес-процесах.

Серверна частина розроблялась на Python з використанням Flask у середовищі PyCharm, побудована за трирівневою архітектурою: рівень представлення, рівень бізнес-логіки та рівень доступу до даних. Клієнтська логіка реалізована за допомогою Jinja2 і CSS-фреймворку Vulma.

Розроблений веб-застосунок сприятиме автоматизації аналізу зворотного зв'язку від клієнтів, зменшуючи витрати часу та ресурсів, що особливо актуально для бізнесу.

Проведено апробацію системи – результати представлені у тезах на XXXVI Міжнародній науковій конференції «iScience», а також подано статтю для публікації.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. John Walker S. Big data: a revolution that will transform how we live, work, and think. *International journal of advertising*. 2014. Vol. 33, no. 1. P. 181–183. URL: <https://doi.org/10.2501/ija-33-1-181-183> (date of access: 20.04.2025).
2. Мелешко Є. В. Алгоритми комп'ютерного аналізу текстів на природній мові : thesis. 2017. URL: <http://dspace.kntu.kr.ua/jspui/handle/123456789/7376> (дата звернення: 20.04.2025).
3. Haykin S. S. Neural networks: A comprehensive foundation. Englewood Cliffs, N.J : Prentice Hall, 2006. 700 p.
4. Tone analytics (Classifications). *IBM Cloud*. URL: https://cloud.ibm.com/docs/natural-language-understanding?topic=natural-language-understanding-tone_analytics (date of access: 20.04.2025).
5. Emotion recognition of morphed facial expressions/ L. C. Jiskoot et al. *Journal of Neurology*. 2020. URL: <https://doi.org/10.1007/s00415-020-10096-y> (date of access: 20.04.2025).
6. Sentiment Analysis with Text Classification Demo. *Python NLTK*. URL: <https://text-processing.com/demo/sentiment/> (date of access: 20.04.2025).
7. Cloud Natural Language | Google Cloud. *Google Cloud*. URL: <https://cloud.google.com/natural-language> (date of access: 20.04.2025).
8. Semantria - Lexalytics. *Lexalytics*. URL: <https://www.lexalytics.com/semantria/> (date of access: 22.04.2025).
9. SQLite documentation. *SQLite Home Page*. URL: <https://www.sqlite.org/docs.html> (date of access: 22.04.2025).
10. Keras: Deep Learning for humans. *Keras*. URL: <https://keras.io/> (date of access: 22.04.2025).

11. Бенджио Й., Гудфеллоу Я., Курвилль А. Глибоке навчання. 2-ге вид. Print2print, 2018. 652 с.
12. Welcome to Flask – Flask Documentation (3.1.x). *Flask*. URL: <https://flask.palletsprojects.com/en/stable/> (date of access: 22.04.2025).
13. Учасники проєктів Вікімедіа. Діаграма прецедентів – Вікіпедія. *Вікіпедія*. URL: https://uk.wikipedia.org/wiki/Діаграма_прецедентів (дата звернення: 22.04.2025).
14. Lucidchart | Diagramming Powered By Intelligence. *Lucidchart*. URL: <https://www.lucidchart.com/pages/> (date of access: 24.04.2025).
15. Lieberherr K. J., Holland I. M. Assuring good style for object-oriented programs. *IEEE software*. 2009. Vol. 6, no. 5. P. 38–48. URL: <https://doi.org/10.1109/52.35588> (date of access: 24.04.2025).
16. Hochreiter S., Schmidhuber J. Long Short-Term Memory. *Neural computation*. 1997. Vol. 9, no. 8. P. 1735–1780. URL: <https://doi.org/10.1162/neco.1997.9.8.1735> (date of access: 24.04.2025).
17. Connectionist architectures for multi-speaker phoneme recognition. *List of Proceedings*. URL: <https://proceedings.neurips.cc/paper/1989/hash/979d472a84804b9f647bc185a877a8b5-Abstract.html> (date of access: 24.04.2025).
18. Graham B. Fractional Max-Pooling. https://www.researchgate.net/publication/269711243_Fractional_Max-Pooling (date of access: 26.04.2025).
19. Moayed H., Mansoori E. G. Improving regularization in deep neural networks by co-adaptation trace detection. *Neural processing letters*. 2023. URL: <https://doi.org/10.1007/s11063-023-11293-2> (date of access: 26.04.2025).
20. Грофф Д., Вайнберг П., Оппель Е. SQL Повний посібник. The McGraw-Hill, 2010. 959 с.
21. GloVe: global vectors for word representation. *The Stanford Natural Language Processing Group*. URL: <https://nlp.stanford.edu/projects/glove/> (date of access: 26.04.2025).

22. Test ASP.NET core MVC apps - .NET. *Microsoft Learn: Build skills that open doors in your career.* URL: <https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/test-asp-net-core-mvc-apps> (date of access: 26.04.2025).