

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ
Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.
_____ Ю. П. Кондратенко
« ____ » _____ 2022 р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ТА КЛАСИФІКАЦІЯ
ТЕКСТІВ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ
ШТУЧНОГО ІНТЕЛЕКТУ

Спеціальність 122 «Комп'ютерні науки»

122 – МКР – 601.21610114

Студент _____ О. О. Малімон
« ____ » _____ 2022 р.

Консультант _____ Є. В. Сіденко
канд. техн. наук, доцент
« ____ » _____ 2022 р.

Миколаїв – 2022

Чорноморський національний університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

Освітньо-кваліфікаційний рівень магістр

Галузь знань 12 «Інформаційні технології»
(шифр і назва)

Спеціальність 122 «Комп'ютерні науки»
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, проф.
_____ Ю. П. Кондратенко
« ____ » _____ 2021р.

ЗАВДАННЯ

на магістерську кваліфікаційну роботу

Малімону Олександрю Олеговичу

(прізвище, ім'я, по батькові)

1. Тема магістерської кваліфікаційної роботи «Інтелектуальний аналіз та класифікація текстів з використанням технологій штучного інтелекту».

Керівник роботи _____ канд. техн. наук, доцент Сіденко Євген Вікторович _____
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затв. наказом Ректора ЧНУ ім. Петра Могили від « ____ » ____ 20__ р. № _____

2. Строк подання студентом роботи « ____ » ____ 20__ р.

3. Вхідні (початкові) дані до роботи: база даних з відгуками, які користувачі вказали для фільмів. База даних зі словами різних за настроєм відтінків _____

Очікуваний результат роботи: розроблена система для аналізу та класифікації тексту за настроєм з високою точністю і високим ступенем довіри, використовуючи нейроні мережі

4. Зміст пояснювальної записки (перелік питань, які потрібно розглянути): дослідження та аналіз сучасного стану задачі інтелектуального аналізу та класифікації текстів, головні типи систем класифікацій, алгоритми класифікації настрою, дослідження впливу різних параметрів на результати _____.

5. Перелік графічних матеріалів презентація

6. Завдання до спеціальної частини опис основних питань охорони праці пов'язаних з професійною діяльністю та використання обрахунків для роботи в офісі

7. Консультанти:

| Розділ | Прізвище, ініціали та посада консультанта | Підпис |
|------------------------------------|---|--------|
| Спеціальна частина з охорони праці | к.т.н., доцент Ю. Г. Щербак | |
| Методична частина | к.т.н., доцент Є. В. Сіденко | |

Керівник роботи канд. техн. наук, доцент Сіденко Є. В.
(наук. ступінь, вчене звання, прізвище та ініціали)

(підпис)

Завдання прийнято до виконання Малімон О. О.
(прізвище та ініціали)

(підпис)

Дата видачі завдання « » _____ 20 р.

КАЛЕНДАРНИЙ ПЛАН

Виконання магістерської кваліфікаційної роботи

Тема: Інтелектуальний аналіз та класифікація текстів з використанням технологій штучного інтелекту

| № | Найменування роботи | Початок | Закінчення | Примітки |
|----|---|------------|------------|----------|
| 1 | Визначення керівника і теми МКР. Подання заяви на затвердження теми МКР | 01.09.2021 | 10.10.2021 | |
| 2 | Отримання завдання на виконання МКР | 19.10.2021 | 22.10.2021 | |
| 3 | Складання календарного плану на період виконання МКР | 23.10.2021 | 26.10.2021 | |
| 4 | Огляд літератури за темою дослідження | 27.10.2021 | 10.11.2021 | |
| 5 | Проходження переддипломної практики, збір та аналіз матеріалів до МКР | 22.11.2021 | 11.12.2021 | |
| 6 | Аналіз предметної області та розробка технічного завдання. Моделювання результатів | 16.12.2021 | 12.01.2022 | |
| 7 | Опис фахової частини МКР, зокрема дослідження публікацій щодо класифікації тексту, огляд існуючих алгоритмів, які вирішують поставлену задачу, реалізація обраних технологій з аналізом отриманих результатів | 13.01.2022 | 25.01.2022 | |
| 8 | Розробка спеціальної частини з охорони праці та методичної частини | 26.01.2022 | 30.01.2022 | |
| 9 | Попередній захист МКР на засіданні комісії кафедри | 31.01.2022 | 31.01.2022 | |
| 10 | Корегування роботи за результатами попереднього захисту | 01.02.2022 | 03.02.2022 | |
| 11 | Остаточне оформлення пояснювальної записки та слайдів доповіді для захисту | 04.02.2022 | 06.02.2022 | |
| 12 | Подання МКР рецензенту | 09.02.2022 | 10.02.2022 | |
| 13 | Рецензування МКР | 11.02.2022 | 12.02.2022 | |
| 14 | Подання МКР, її електронної копії та інших документів (відгуку, рецензії) до захисту | 14.02.2022 | 15.02.2022 | |
| 15 | Захист МКР перед екзаменаційною комісією (ЕК) | 21.02.2022 | 22.02.2022 | |

Розробив студент Малімон Олександр Олегович
(прізвище та ініціали) (підпис)

Керівник роботи канд. техн. наук, доцент Сіденко Євген Вікторович
(наук. ступінь, вчене звання, прізвище та ініціали) (підпис)

«23» жовтня 2021 р.

АНОТАЦІЯ

до магістерської кваліфікаційної роботи
студента групи 601 ЧНУ ім. Петра Могили

Малімона Олександра Олеговича

на тему: **“ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ТА КЛАСИФІКАЦІЯ ТЕКСТІВ З
ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ ШТУЧНОГО ІНТЕЛЕКТУ”**

Актуальність даного дослідження полягає у необхідності підвищення ефективності класифікації і аналізу текстів із визначенням настрою користувача, розробці програмного забезпечення з використанням сучасних методів для вирішення відповідної задачі. Це дозволить зменшити час обробки відгуків та зауважень користувачів і, відповідно, автоматизувати даний процес.

Об’єктом дослідження є процеси аналізу та класифікації текстів.

Предметом дослідження є методи інтелектуального аналізу та класифікації текстів.

Метою роботи є інтелектуальний аналіз та класифікації текстів для зменшення часу, який витрачається на обробки відгуків та зауважень користувачів, а також підвищення якості обслуговування клієнтів за рахунок автоматизації даного виду робіт.

В результаті виконання роботи було досліджено два методи класифікації (метод Naive Bayes і штучна нейрона мережа), проаналізовано вплив їх внутрішніх параметрів на роботу алгоритмів, визначені основні їх переваги та недоліки, а також розроблено програмне забезпечення, в якому реалізовані відповідні методи.

Дана робота складається з шести розділів. Кожен розділ відповідно присвячений: аналізу предметної області, математичним моделям і методам, використаним у магістерській роботі, моделюванню і проектуванню системи планування та оптимізації маршрутів, аналізу отриманих результатів, охороні праці, методичній частині. Загальний обсяг роботи – 104 сторінки. Кваліфікаційна робота містить: 64 рисунки, 1 таблицю і посилання на 54 літературних джерела.

Ключові слова: класифікація тексту, аналіз настроїв, метод Naive Bayes, нейрона мережа.

ABSTRACT

to the master's qualification work by the student of the group 601 of Petro Mohyla Black Sea National University

Malimon Oleksandr

“INTELLECTUAL ANALYSIS AND CLASSIFICATION OF TEXTS USING ARTIFICIAL INTELLIGENCE TECHNOLOGIES”

The relevance of this study is the need to improve the efficiency of classification and analysis of texts to determine the mood of the user, software development using modern methods to solve the problem. This will reduce the processing time of user feedback and comments and also automate the process.

The object of research is the processes of analysis and classification of texts.

The subject of research is methods of intellectual analysis and classification of texts.

The aim of the work is the intellectual analysis and classification of texts to reduce the time spent on processing feedback and comments from users, as well as improving the quality of customer service by automating this type of work.

As a result of the work, two classification methods were investigated (Naive Bayes method and artificial neural network), the influence of their internal parameters on the operation of algorithms was analyzed, their main advantages and disadvantages were identified, and software was developed.

This work consists of six sections. Each section is devoted to: analysis of the subject area, mathematical models and methods used in the master's thesis, modeling and design of planning and optimization of routes, analysis of results, labor protection, methodological part of the master's thesis. The overall scope of the work is 104 pages. Thesis contains 64 figures, 1 table and 54 sources in it.

Key words: text classification, mood analysis, Naive Bayes method, neural network.

ЗМІСТ

| | |
|---|----|
| ПЕРЕЛІК СКОРОЧЕНЬ..... | 3 |
| ВСТУП..... | 6 |
| 1 АНАЛІЗ СУЧАСНОГО СТАНУ ЗАДАЧІ КЛАСИФІКАЦІЇ ТЕКСТУ | 7 |
| 1.1 Аналіз предметної сфери..... | 14 |
| 1.2 Останні дослідження та публікації..... | 18 |
| 1.3 Огляд існуючих аналогів..... | 21 |
| 1.4 Огляд недоліків та можливі їх покращення | 25 |
| 2 ТЕХНОЛОГІЇ ТА АЛГОРИТМИ КЛАСИФІКАЦІЇ ТЕКСТУ | 29 |
| 2.1 Нейронні мережі..... | 29 |
| 2.2 Naive Bayes | 35 |
| 2.3 Порівняльний аналіз вищеперерахованих методів..... | 41 |
| 3 РЕАЛІЗАЦІЯ СИСТЕМИ АНАЛІЗУ ТА КЛАСИФІКАЦІЇ ТЕКСТУ | 42 |
| 3.1 Опис технологій | 42 |
| 3.2 Опис Web API..... | 48 |
| 3.3 Опис веб-застосунку | 50 |
| 3.4 Опис плагіну нейронної мережі | 51 |
| 3.5 Опис плагіну Naive-bayes..... | 52 |
| 4 ОПИС РОЗРОБЛЕНОЇ СИСТЕМИ ТА АНАЛІЗ РЕЗУЛЬТАТІВ | 54 |
| 4.1 Опис застосунку | 54 |
| 4.2 Опис діаграм | 57 |
| ВИСНОВКИ..... | 77 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ..... | 79 |

ПЕРЕЛІК СКОРОЧЕНЬ

AOT — Ahead-of-time.

API — Application Programming Interface.

CLI – Command Line Interface.

CLR — Common Language Runtime.

CS — Computer Science.

CSS — Cascade Style Sheet.

CSV — Comma Separated Values.

GC — Garbage Collector.

HTML — Hypertext Markup Language.

IL — Intermediate Language.

IT — Information Technology.

JSON — JavaScript Object Notation.

ML — Machine Learning.

MVC — Model-View-Controller.

RX — Reactive extensions.

SDK — Software Development Kit.

SPA – Single Page Application.

Пояснювальна записка

до магістерської кваліфікаційної роботи

на тему:

«ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ТА КЛАСИФІКАЦІЯ ТЕКСТІВ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ ШТУЧНОГО ІНТЕЛЕКТУ»

Спеціальність 122 «Комп'ютерні науки»

122 – МКР – 601.21610114

Студент _____ О. О. Малімон
«__» _____ 2022р.

Консультант _____ Є. В. Сіденко
канд. техн. наук, доцент
«__» _____ 2022р.

Миколаїв – 2022

ВСТУП

В сучасному світі більшість людей має власні профілі у соціальних мережах, які використовуються для висловлення власної думки. З всього цього може отримати користь бізнес, який за допомогою аналізу відгуків зможе отримати об'єктивну думку згідно своїх товарів або послуг. Це призвело до розвитку алгоритмів й систем, які б могли класифікувати й аналізувати текст.

За допомогою систем аналізу тексту, власники бізнесу могли б швидше обробляти коментарі й відгуки користувачів. Все це призвело б до покращення розуміння користувачів власних сервісів, їх побажань й причин їх незадоволення. Перевагою таких систем є те, що їх можна використовувати у багатьох сферах, де люди можуть ділитись власними думками. Найбільш популярним прикладом є інтернет-магазини та соціальні мережі.

Потреба у системах аналізу тексту, в наш час, є занадто великою адже вони роблять можливим автоматизації значної частини роботи, яку раніше люди й витрачали багато часу на це. Тож, як результат, великі ІТ-компанії активно використовують їх у своїх програмах. Найвідоміші системи класифікації й аналізу тексту були створені й інтегровані у свої програми такими компаніями як Facebook, Twitter, MonkeyLearn, Repustate та іншими.

Як результат, системи класифікації й аналізу тексту є дуже затребуваними в наш час, адже за їх допомогою, обробка відгуків стає автоматизованою. Так як результатом виконання магістерської роботи буде клієнтський застосунок, який буде містити коментарі користувачів на платформі IMDB, то він буде затребуваним серед усіх компаній які розроблюють власні сервіси та продукти. За допомогою розробленої системи можна буде отримувати інформацію, про настрої користувача, адже результати базувались би на його відгуках та коментарях.

Метою роботи є вдосконалення існуючого процесу аналізу тексту, шляхом створення класифікаційної системи та клієнтського застосунку.

1 АНАЛІЗ СУЧАСНОГО СТАНУ ЗАДАЧІ КЛАСИФІКАЦІЇ ТЕКСТУ

В сучасному світі більшість людей має власні профілі у соціальних мережах. Багато хто використовує свої сторінки висловлення власної думки. З всього цього може отримати користь бізнес, який за допомогою аналізу відгуків та коментарів в різних спільнотах зможе отримати об'єктивну та обґрунтовану думку згідно своїх товарів або послуг. Кожен бізнесмен хоче якомога краще розуміти клієнтів, їх настрої, потреби та думки – це дозволить покращити якість продукції та збільшити базу покупців.

В ролі великих компаній, які займаються розробкою веб-проектів, якими користуються мільйони людей по всьому світу, необхідно виконувати аналіз сотні тисяч або навіть мільйонів відгуків та думок згідно власних проектів й банальний пошук слів “добре” та “погано” в коментарях не є достатнім [10]. З стрімким розвитком алгоритмів машинного навчання, а саме нейронних мереж, проблему аналізу емоцій, які користувач намагався передати в повідомленні стало можливо вирішувати з високою точністю.

Задачею аналізу тексту є класифікація полярності тексту, речення, або ж документу – визначення того наскільки виражена думка є позитивною, негативною чи нейтральною. Розвитком даних алгоритмів є пошук джерела або контексту, коли алгоритм сам визначає не тільки емоції користувача але й про що він пише. Приклад аналізу речень можна побачити на рисунку 1.1.

На цьому рисунку можна побачити три різні відгуки й їх відношення до класів “позитивне”, “негативне” та “нейтральне”.

Як приклад[7], уявімо, що користувач написав відгук який виглядає наступним чином: “Я впродовж тривалого часу користувався вашим сервісом XXXXX й не був засмученим його якістю. Можу зазначити, що більшість користувачів недооцінили якість цього сервісу”. Власники компанії хочуть проаналізувати відгуки й визначити,

чи користувачі задоволені сервісом XXXXX. Нажаль, більшість систем, які базуються базі правил, або ж ключових слів не знайдуть таких слів як “задоволений”, “щасливий” або “гарний”.

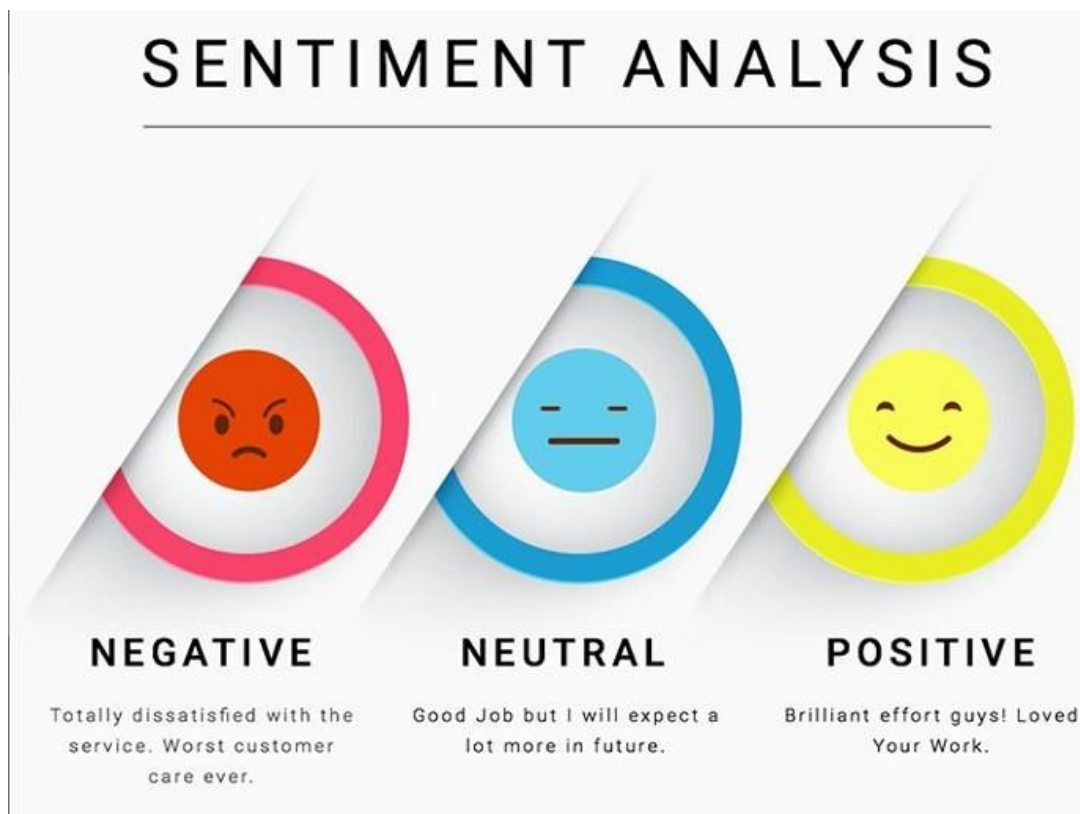


Рис. 1.1. Визначення емоцій користувача в відгуках

Навпаки будуть виділені такі слова як “засмучений” та “недооцінили”, які призведуть до некоректної класифікації речення й результат буде нейтральним або навіть негативним. Сучасні підходи, з використанням нейронних мереж можуть коректно впоратись з даною проблемою.

Аналізом настроїв користувачів є контекстним аналізом тексту, який визначає та витягує суб’єктивну інформацію з вихідного матеріалу та допомагає компанії зрозуміти соціальні настрої свого бренду, продукту чи послуги під час моніторингу онлайн-розмов. Однак аналіз потоків соціальних мереж зазвичай обмежується лише

основним аналізом настроїв і показниками на основі підрахунку. Це схоже на те, щоб просто зірвати поверхню й упустити ці цінні ідеї, які чекають на відкриття. Отже, що повинен зробити бренд, щоб захопити цю сферу?

З нещодавнім розвитком алгоритмів машинного навчання, можливість алгоритмів для аналізу тексту значно виросла. Креативне використання прогресивних методів штучного інтелекту може бути ефективним застосунком для виконання досліджень. Важливо аналізувати розмови користувачів про бренд адже це допомагає визначити:

- 1) ключові аспекти бренду, продукту та сервісу, якими користувачі є зацікавленими;
- 2) реакції користувачів згідно намірів на зміни в цих аспектах.

Ці базові концепції, при поєднанні, стають дуже важливими застосунками для аналізу мільйону розмов користувачів про брендів. Класифікація тексту складається з базових та найбільш важливих кроків:

1. **Аналіз настроїв** є найбільш популярним застосунком, який класифікує всі вхідні повідомлення та вказує на те, яке з них є позитивним, негативним або ж взагалі нейтральним;

2. **Аналізатор намірів** є більше важливим й вагомим типом алгоритмів класифікаторів. За їх допомогою можливо аналізувати наміри, які користувачі намагались виразити в повідомленні та визначення чи відносяться ті наміри до думок, новин, реклами, скарги, пропозиції, запиту чи вираженням поваги. На рисунку 1.2 можна побачити приклад того, як речення, яке користувач опублікував в соціальній мережі Twitter аналізується й згодом класифікується як запит. На цьому рисунку можна побачити, як повідомлення, яке було опубліковане користувачем, аналізується й згодом класифікується як запит, який потребує подальшого аналізу та взаємодії з людиною;

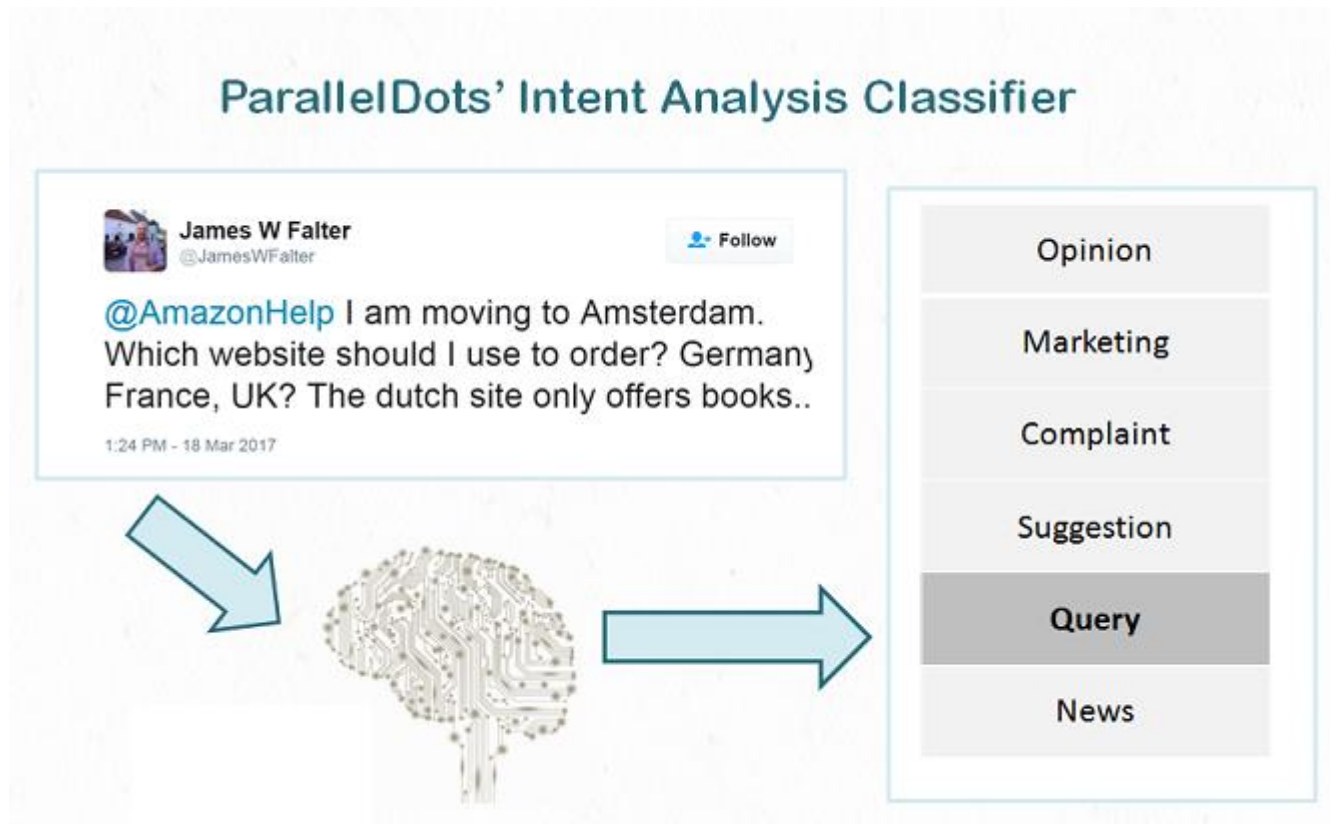


Рис. 1.2. Класифікація повідомлення користувача

3. Контекстний семантичний пошук. Це є найбільш цікавим кроком. Для того, щоб визначити найбільш необхідні рішення та кроки необхідні, необхідно зрозуміти про які аспекти бренду користувачі розмовляють. Наприклад, Amazon хоче розділити повідомлення на різні категорії: пізні доставки, проблеми з оплатою, запити на підвищення по посаді, огляди продуктів та інші. У той же час така компанія як Starbucks хоче класифікувати повідомлення по іншим категоріям, а саме: чи відносяться повідомлення поведінки персоналу, нових смаків кави, рівню гігієни в приміщеннях, онлайн замовлень, назв магазинів їх місцезнаходження та інше. Всі ці питання та проблеми можливо вирішувати за допомогою контекстного семантичного пошуку. Спосіб, за яким це працює: необхідно брати тисячі повідомлень й концепцію або ж тему ціни. Використовувати повідомлення як вхідні дані й фільтрувати ті які хоча б трохи схожі за темою. Далі, на рисунку 1.3 можливо

детально побачити як працює контекстний семантичний пошук.

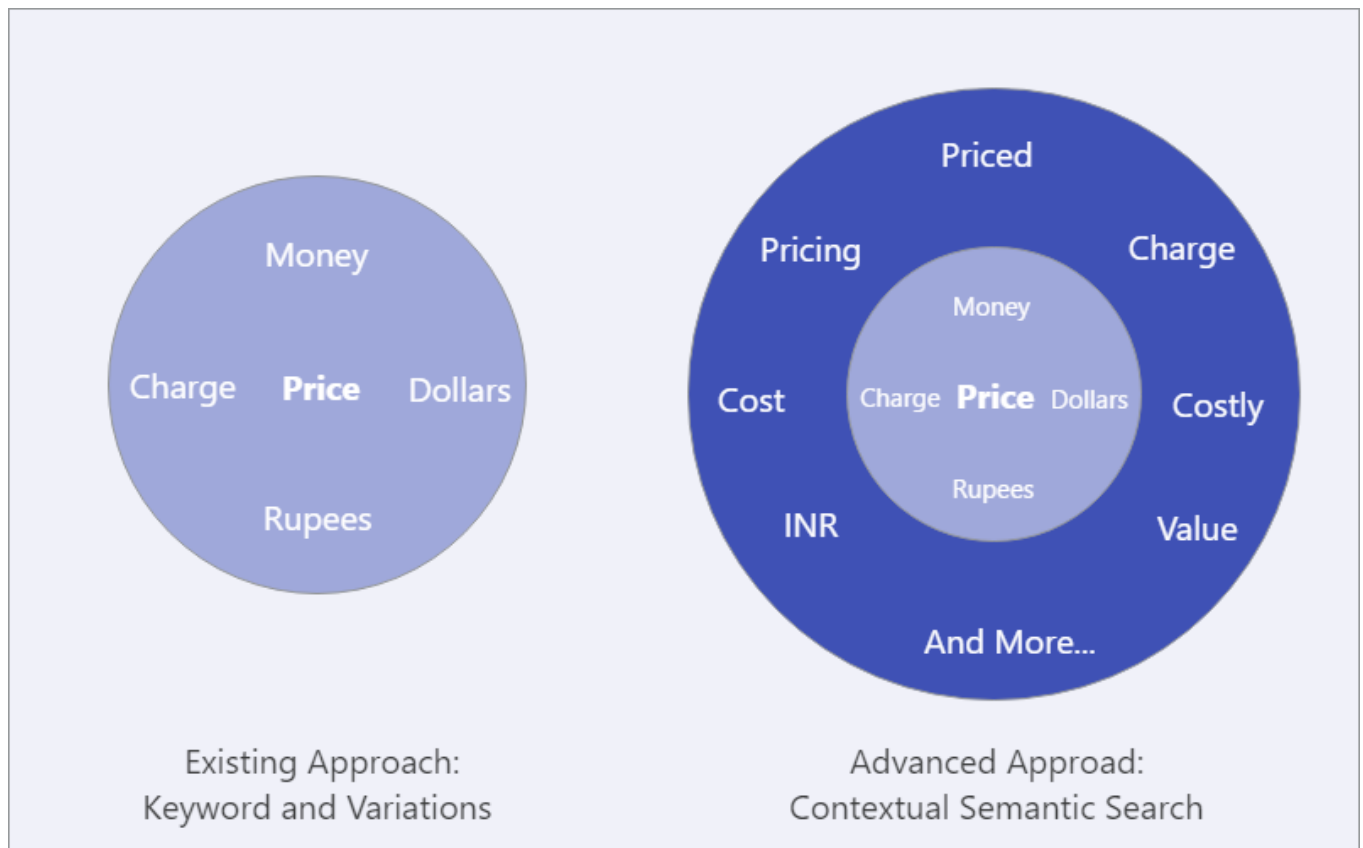


Рис. 1.3. Приклад роботи контекстного семантичного пошуку

На цьому рисунку можна побачити, як працює звичайний алгоритм пошуку по ключовим словам й те як працює контекстний семантичний пошук. Очевидно, що другий спосіб є кращим адже він визначає значно більше повідомлень, які відносять до теми ціни ніж перший.

Звичайні методи та підходи для фільтрації по темі ціни працюють наступним чином: шукають всі слова які схожі на ціну або відносяться до цієї теми (ціна, оплата, \$, виплати та ін.). Такий метод є не занадто ефективним, адже неможливо визначити та створити такий словник, який буде містити абсолютно всі ключові слова, для головних тем. В той же час контекстний семантичний пошук в якості вхідного параметру використовує лише слово “ціна” й залишає майже всі

повідомлення, навіть ті, які не містять ключових слів у власному тексті.

Далі буде наведено приклад для того, щоб зрозуміти детальніше як це працює. За допомогою використання методів штучного інтелекту вдалося перетворити слова в точки в конкретній позиції в просторі. Дистанція між цими точками використовується для визначення схожості повідомлень за контекстом, який є метою пошуку. Візуалізацію того, як це виглядає можна побачити на рисунку 1.4.



Рис. 1.4. Візуалізація повідомлень в соціальній мережі Twitter

На цьому рисунку можна побачити, як виглядає візуалізація повідомлень в мережі Twitter, які є схожими за конкретною сферою та контекстом.

Час подивитись на реальних прикладах, як працює контекстний семантичний пошук при аналізі повідомлень в різних соціальних мережах. На рисунку 1.5 можна побачити перший приклад повідомлення про компанію Uber, який було опубліковано в соціальній мережі Facebook.

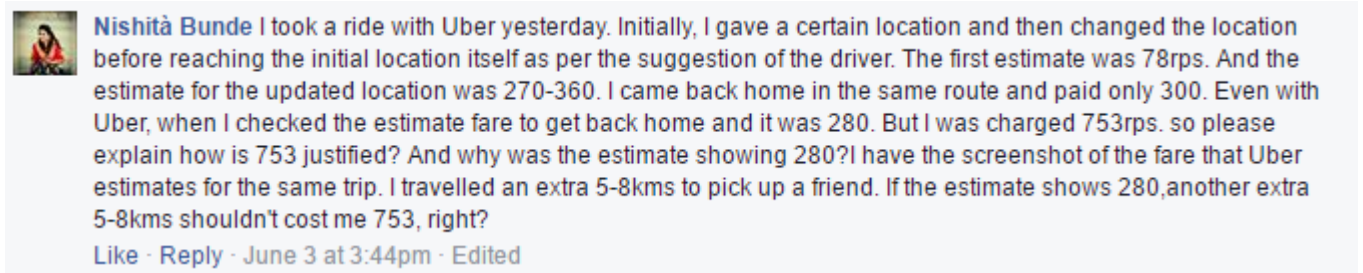


Рис. 1.5. Приклад повідомлення в соціальній мережі Facebook

На цьому рисунку можна побачити приклад повідомлення в соціальній мережі Facebook, яке відноситься до компанії Uber.

На рисунку 1.6 можна побачити другий приклад повідомлення про компанію Uber, що відноситься до категорії ціни, який було опубліковано в соціальній мережі Twitter.

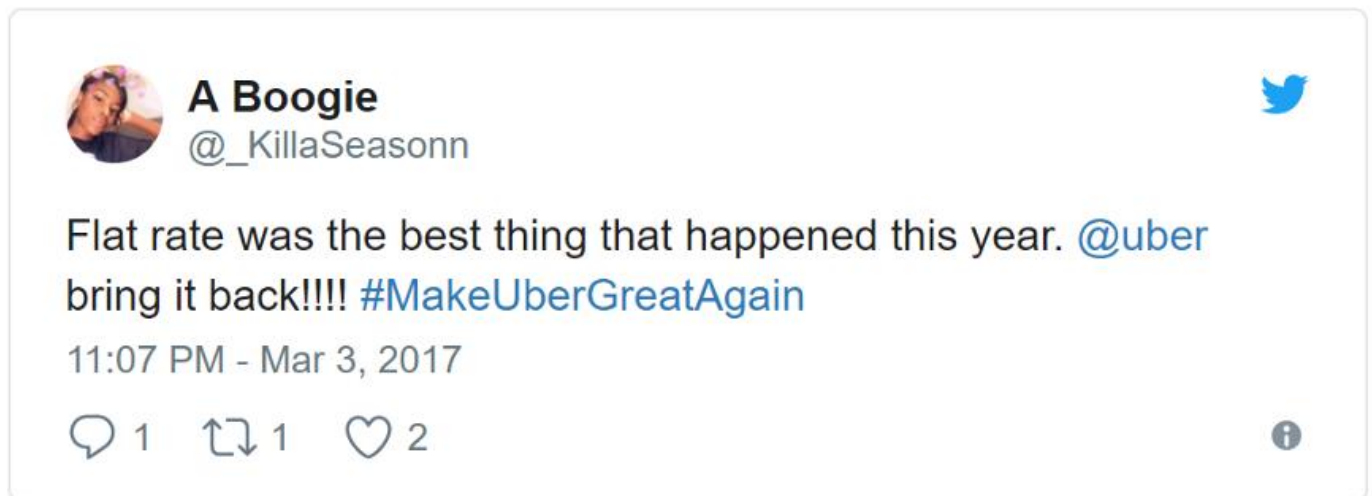


Рис. 1.6. Приклад повідомлення в соціальній мережі Twitter

На цьому рисунку можна побачити приклад повідомлення в соціальній мережі Twitter, яке відноситься до компанії Uber.

В обох прикладах, а саме на рисунку 1.5 та рисунку 1.6, вони будуть визначені алгоритмом, як ті, що відносяться за тематикою до питань ціни, навіть не зважаючи

на те, що дане слово не зустрічається у обох повідомленнях.

Методи класифікації настроїв можливо робити за допомогою наступних алгоритмів: машинного навчання, лексичних правил та гібридний підходи. Машинне навчання використовує лінгвістичні особливості. Лінгвістичний підхід базується на колекціях слів відомих слів. Цей підхід використовує словники, які використовують статистичні або семантичні методи для визначення полярності настроїв. Гібридний підхід використовує одночасно два попередніх й є найбільш популярним під час вирішення даної проблеми.

Класифікація тексту за допомогою нейронних мереж може мати навчання двох типів: з вчителем та самостійна. Методи навчання з вчителем оброблюють велику кількість даних для тестування. Методи без вчителя використовуються тоді, коли важко визначити ці дані для тренування. Уявімо, що є набір даних для навчання де кожний запис складається з полів X_1 , X_2 й X_n . Модель для класифікації базується на особливостях, які є в записі. Методи, які потребують вчителя, залежать від існування результуючого поля в наборі даних.

Слова та думки використовуються в багатьох завданнях класифікації настроїв. Слова позитивні використовуються для вираження деяких бажань, тоді як слова негативні для вираження небажаних. Існує два основних підходи для складання списку таких слів: ручний та автоматизований. Цей підхід використовує вже створені словники, які були створені за допомогою попередніх методів. У даних методів є значний недолік – неможливість визначати контекст та специфіку речення.

1.1 Аналіз предметної сфери

Алгоритми аналіз тексту – це методи які використовуються для оцінки думок людей або груп, наприклад, сегменту аудиторії якогось бренду. За допомогою механізму підрахунку балів алгоритми визначають ставлення, думки та емоції пов'язані з якоюсь подією, послугою чи темою, які відчуває людина. Аналіз настроїв

іноді називають аналізом думок. Аналіз тексту є невід'ємним компонентом системи аналітики мовлення, який визначає думку або ставлення клієнту.

Аналіз тексту частіш за все використовує алгоритми та підраховують кількість слів, які вказують на почуття людини. Алгоритми можуть використовувати більшу кількість показників [2, 11] для класифікації, такі як:

- 1) як швидко людина говорить (темп мовлення);
- 2) зміни рівня стресу, про які свідчить промова людини;
- 3) рівень стресу або розчарування в голосі.

В реальному світі алгоритми аналізу тексту використовуються в багатьох застосунках й можуть бути застосовані для різних цілей. Як приклад, можуть бути використані для аналізу текстів в соціальній мережі Twitter для отримання зваженої позиції користувачів згідно якоїсь конкретної події. Великі компанії часто використовують алгоритми аналізу тексту для розуміння ставлення користувачів по всьому світу до власних товарів.

Алгоритми класифікації тексту [4, 9] використовуються для таких цілей як:

- 1) за допомогою перевірки коментарів, організації можуть брати до уваги лише відгуки від тих користувачів, які система буде вважати адекватними;
- 2) для врахування думок користувачів згідно послуг які ви надаєте й їх можливих покращень;
- 3) для врахування негативних думок, де люди можуть вказувати на недоліки які необхідно виправити;
- 4) для постійного моніторингу відгуків з метою переконання в тому, що зміни в ваших сервісах мають позитивний результат і кількість негативних коментарів зменшується.

Алгоритми, які використовують [5-6] під час розборки систем для класифікації тексту:

- 1) машинне навчання – це підхід, за допомогою якого можливо визначити

особливості та створити класифікатор, який зможе тип емоцій які виражені в тексті;

2) lexicon-based – це підхід, де використовуються набори слів які виражають різні емоції для класифікації тексту;

3) гібридний – це підхід, який використовує машинне навчання в комбінації з lexicon-based способом.

В наш час великі й малі компанії використовують алгоритми класифікації тексту для оцінки ставлення користувачів власних продуктів та систем згідно їхньої якості. Робиться це також для отримання в реальному часі думок користувачів, їх уподобань та досвіду використання власних сервісів та застосунків.

Використання даних алгоритмів дозволяє вам, як власнику, або ж менеджеру, визначати моментально в реальному часі користувачів, які сумніваються, або щось не розуміють, а також незадоволених. Після виявлення таких людей, у вас з'являється можливість взаємодіяти з ними, зв'язатись з метою визначення причини їх незадоволення. В подальшій перспективі, якщо наявний відділ підтримки користувачів, то це значно полегшить їх роботу, а саме: їм не знадобиться витратити більше час на перечитування всіх коментарів з метою пошуку незадоволених користувачів, або ж тих, хто потребує допомоги з використанням сервісів. Все призведе до значного покращення якості сервісів, більшої продуктивності деяких конкретних відділів з працівниками й їх задоволеності разом з користувачами. Можливість визначати кризисну ситуацію швидко дозволить вам прийняти моментальні рішення, які допоможуть зберегти репутацію вашого бренду.

Як менеджер або власник сервісу чи застосунку ви матимете також змогу визначати реакцію користувачів на якісь специфічні події, такі як запуск нової маркетингової компанії. Використовуючи інструменти аналізу настроїв на основі аспектів, ви можете охоплювати пені теми в соціальних мережах. Такий підхід дозволяє дізнатися, що клієнтам подобається і що не подобається у вашому продукті: що вони думають про ваше обслуговування клієнтів? Які їхні улюблені функції.

Згідно оцінок, очікування відповіді є причиною №1 серед користувачів, чому вони є незадоволеними сервісом. Один із способів для бізнесу, який може покращити час відповіді є автоматизація визначення теми та позначення повідомлень, які надходять до сервісу надання допомоги, специфічними тегами.

Ваш сервіс взаємодії з користувачами також може використовувати алгоритми аналізу тексту для позначення вхідних запитів від користувачів базуючись на їх емоційному забарвленні. Після такого визначення, повідомлення з негативним забарвленням будуть мати більший пріоритет й відображатись першими в черзі. Таким способом, ви будете впевненими, що найбільш термінові повідомлення будуть обробленими в першу чергу. В той час ви в праві створити правила для автоматичного направлення повідомлень в відповідний відділ, який буде займатись їм, або ж до специфічних робітників. Як приклад, повідомлення від незадоволених користувачів будуть направленні до найбільш кваліфікованого працівника в команді, для того, щоб бути максимально впевненим, що користувач отримає гідну допомогу та коректне ставлення.

Аналіз тексту може допомогти бізнесу визначити тренди та вивчити нові можливості на ринку. Якщо ж ви плануєте запускити новий продукт, вам захочеться відстежувати ставлення користувачів до нього в соціальних мережах. Це є чудовим способом для отримання розуміння згідного того, що ваш користувач очікує та потребує. У схожому ж способі ви можете використовувати аналіз речень для оцінки відгуків ваших конкурентів, для того щоб зрозуміти чого користувачі очікують від інших продуктів та сервісів. Що робиться як треба? Які слабкі моменти у них є? Використовуючи цю інформацію ви можете побудувати специфічну стратегію ведення власного бізнесу, яка б брала до уваги всі, або хоча б більшість, недоліків.

Підводячи підсумки, за допомогою алгоритмів аналізу тексту дані, які ви отримуєте від користувачів стають вагомою перевагою в порівнянні з іншими компаніями, які такими системами не користуються. За допомогою таких систем ви

можете зрозуміти ставлення користувачів до власного бренду, системи, застосування та іншого.

1.2 Останні дослідження та публікації

Завдяки швидкому розвитку та популяризації онлайн-магазинів, все більше користувачів роблять покупки на різних платформах. Аналізуючи тенденцію оцінювань настроїв споживачів, можна не лише отримати орієнтир для інших споживачів але також покращити якість обслуговування то послуг. Аналіз настроїв який базується на оглядах товарів, також відомий як аналіз тексту, відноситься до процесу автоматичного аналізу суб'єктивного коментаря з емоційним забарвленням замовника і визначення емоційної тенденції клієнта.

Перша публікація [3] називається “Sentiment Analysis for E-Commerce Product Reviews in Chinese Based on Sentiment Lexicon and Deep Learning” була опублікована в 2020 році такими авторами як Li Yang, Ying Li, Jin Wang та R. Simon Sherratt. В цій публікації автори пропонують власну реалізацію SLCABG моделі для класифікації тексту яка базується на перевагах лексичного аналізу та технологіях глибокого навчання. Розроблена система використовує всі переваги лексичного аналізу, CNN та GRU. Під час розробки використовувались дані з реального сайту для продажу книг. Також було вивчено вплив таких параметрів як розмір набору даних, довжина речення та вплив кількості ітерацій на ефективність роботи системи. Система складається з шести слоїв, детальну схему взаємодії між ними можна побачити на рисунку 1.7.

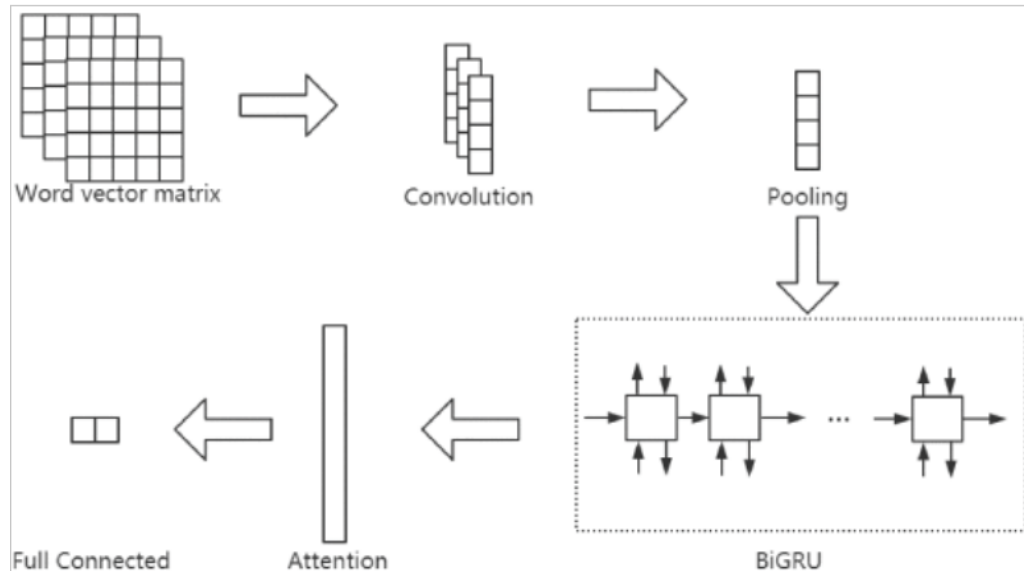


Рис. 1.7. Процес обробки даних в системі

На цьому рисунку можна побачити процес обробки даних в системі й взаємодії між слоями.

Далі автори виконали порівняльну характеристику ефективності власної системи з іншими (рисунок 1.8).

| model | accuracy | precision | recall | F1 |
|----------------------|----------|-----------|--------|-------|
| NaiveBayes [58] | 57.9% | 55.6% | 79.2% | 65.3% |
| SVM [36] | 67.7% | 93.8% | 38.4% | 54.5% |
| CNN [59] | 90.9% | 91% | 90.2% | 90.6% |
| CNN+Attention [60] | 91.4% | 90.8% | 91.6% | 91.2% |
| BiGRU [61] | 92.6% | 91.1% | 94.1% | 92.6% |
| BiGRU+Attention [62] | 93.1% | 92.8% | 93.2% | 93% |
| SLCABG (Ours) | 93.5% | 93% | 93.6% | 93.3% |

Рис. 1.8. Порівняльна таблиця алгоритмів

На цьому рисунку можна побачити порівняльну характеристику різних алгоритмів для аналізу тексту де видно, що точність системи складає 93.5% й є найкращим результатом серед усіх.

Наступна публікація [1] називається “Deep Convolution Neural Networks for Twitter Sentiment Analysis” була опублікована в 2018 році такими авторами як Zhao Jianqiang, Gui Xiaolin та Zhang Xuejun. В даній публікації автори використовують алгоритм згортки для навчання нейронної мережі для покращення точності й швидкості аналізу. Під час навчання було використано дані з соціальної мережи Twitter. Дана система вміє класифікувати коментарі лише як позитивні або негативні. Далі на рисунку 1.9 зображено схема обробки тексту повідомлення.

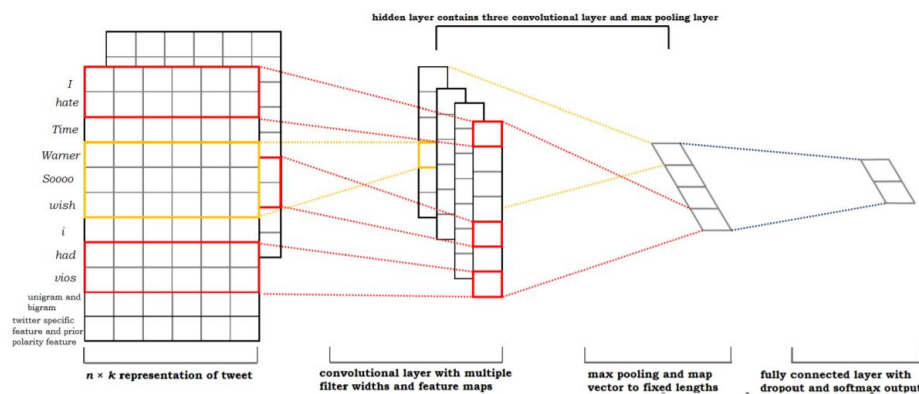


Рис. 1.9. Схема обробки тексту повідомлення

На цьому рисунку можна побачити процес обробки тексту повідомлень й отримання результатів.

Після розробки автори порівняли свою систему з іншими (рисунок 1.10) використовуючи п'ять різних наборів даних й побачили, що точність складає приблизно 87.62%.

| Method | Accuracy /% | | | | | |
|----------------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | SSTd | SED | STSGd | SE2014 | STSTd | Average |
| BoW-SVM | 59.11 | 62.51 | 68.79 | 73.21 | 68.81 | 66.49 |
| BoW-LR | 69.64 | 80.1 | 75.86 | 73.98 | 71.04 | 74.12 |
| GloVe-SVM | 79.16 | 85.87 | 80.07 | 83.06 | 81.61 | 81.95 |
| GloVe-LR | 77.11 | 86.72 | 80.17 | 82.55 | 81.62 | 81.63 |
| GloVe-DCNN | 81.36 | 87.39 | 85.97 | 85.82 | 87.62 | 85.63 |
| SentiStrength[46] | - | - | 81.32 | - | - | - |
| MaxEnt[38] | - | - | - | - | 83.0 | - |
| Hbd[45] | - | - | 80.33 | - | - | - |
| Updated+Expanded[47] | - | - | 82.3 | - | - | - |
| LProp[44] | - | - | - | - | 84.7 | - |
| CharSCNN[16] | - | - | - | - | 86.4 | - |

Рис. 1.10. Порівняльна таблиця алгоритмів

На цьому рисунку можна побачити порівняльну характеристику різних алгоритмів для аналізу тексту де видно, що точність системи складає 87.62% й є найкращим результатом серед усіх.

1.3 Огляд існуючих аналогів

Аналіз настроїв – автоматизований процес розпізнавання позитивних, негативних й нейтральних емоцій в текстових даних, які публікують користувачі. Аналізуючи настрої в коментарях у соціальних мережах, оглядах та відповідях на опитування, компанії можуть краще зрозуміти, як клієнти ставляться до своїх брендів і продуктів. Нижче буде наведено приклади найбільш успішних компаній, які використовують у власних продуктах або ж розроблюють системи для класифікації та аналізу тексту.

Алгоритми аналізу тексту допомагають бізнесу краще зрозуміти настрої та відношення користувача до власного бренду, які надають їм інформацію щодо покращення власних сервісів, застосунків, тощо. Це робить можливим покращення вже існуючих сервісів, що призведе до підвищення кількості клієнтів та більшої задоволеності вже існуючих. В той же час аналіз та класифікація тексту привносить автоматизації в рутині процеси на роботі й дозволяє сконцентруватись на більш

важливих задачах, базуючись на вже отриманих даних про відношення користувачів до власних продуктів.

MonkeyLearn – це компанія SaaS типу, надає програмне забезпечення як головну послугу. Головним продуктом є система класифікації та аналізу тексту за допомогою машинного навчання. В даному продукті вже існують стандартні моделі, які є попередньо реалізовані, їх можна одразу запустити й легко інтегрувати в існуючі власні застосунки, а також ж популярними сторонніми програмами, такими як Google Sheets, Zapier або Zendesk. Користувачі можуть використовувати API, яке підтримує всі основні та найбільш популярні мови програмування і надають ще більшу гнучкість під час підключення інструментів. У випадку, якщо вам потрібні більш точні результати аналізу, користувач може створити індивідуальну модель аналізу настроїв і навчити її за допомогою власних даних. Таким чином система зможе виявити специфічний для конкретного домену словник у ваших заявках у службу підтримки клієнтів, даних соціальних мереж або відповідях на опитування. Процес навчання швидкий, інтуїтивний та зрозумілий. Найкращим є те, що досвід в програмуванні або ж машинному навчанні взагалі не є потрібним.

Repustate – компанія, яка займається розробкою сервісів, які займаються аналізом настроїв в тексті й пропонує API, який підтримує текст на 24 мовах та надає достовірну інформації з даних соціальних мереж. Їх програмне забезпечення поєднує семантичний аналіз, щоб виявити емоції, виражені в емодзі, сленгу, аббревіатурах і хештегах. Створюючи індивідуальні правила аналізу настроїв, користувач може адаптувати свою модель до певного галузевого словника. Нарешті, також можливо інтегрувати API у свій робочий процес за допомогою клієнтський бібліотек, доступних на різних мовах програмування.

Lexalytics – компанія, яка надає рішення для аналізу настроїв з 2003 року. Вони пропонують локальне рішення (Salience) і хмарний API (Semantria), які ви можете інтегрувати у свої робочі процеси. Також можливо створювати повністю

налаштовані моделі та адаптувати їх до потреб вашого бізнесу. Є можливість створювати інформаційні панелі та візуалізації зі своїх даних і отримати більш глибоке уявлення про емоції клієнтів, які використовують ваш сервіс.

Rapidminer – це платформа для науки про дані, яка допомагає підприємствам створювати прогностні моделі на основі даних. Використовуючи Auto Model і Rapidminer Go ви можете використовувати попередньо створені моделі для аналізу настроїв і налаштовувати їх без коду. Платформа інтуїтивно зрозуміла та добре задокументована. Однак знання машинного навчання настійно рекомендується оскільки це дозволить вам вирішувати більш складні завдання.

Lionbridge – компанія, яка надає послуги анотації настроїв для компаній, які хочуть створювати моделі аналізу настроїв за допомогою машинного навчання. Маючи розгалужену мережу учасників по всьому світу, вони можуть вручну додавати теги до тексту більш ніж 30 мовами. Можливе налаштування рішення, виходячи з обсягу вашого проекту та графіку. Після цього необхідно просто завантажити свої дані та визначити параметри для позначення ваших даних.

Sentiment Analyzer – безкоштовний інструмент, який є надзвичайно простим у використанні. Просто необхідно скопіювати та вставити текст, після чого натиснути на кнопку “Аналізувати текст” і вашим результатам буде присвоєно оцінку настрою. Оцінка настрою -100 вказує на надто негативний тон, тоді як оцінка +100 відповідає дуже позитивному.

Uber – найбільш цінний стартап у світі. Працюючи більш ніж у 500 містах по всьому світу й обслуговуючи гігантську базу користувачів, Uber отримує велику кількість відгуків, пропозицій й скарг від людей. Часто соціальні мережі є більш задовільним способом виразити власну думку. Велика кількість вхідних даних робить аналіз, категоризацію й генерацію рішень складним процесом. Аналізуючи більшість популярних соціальних мереж було отримано, що в Facebook – 34173 повідомлення, в Twitter – 21603, та в новинах – 4245. Результати розподілу

коментарів про компанію Uber можна побачити на рисуюнок 1.11.

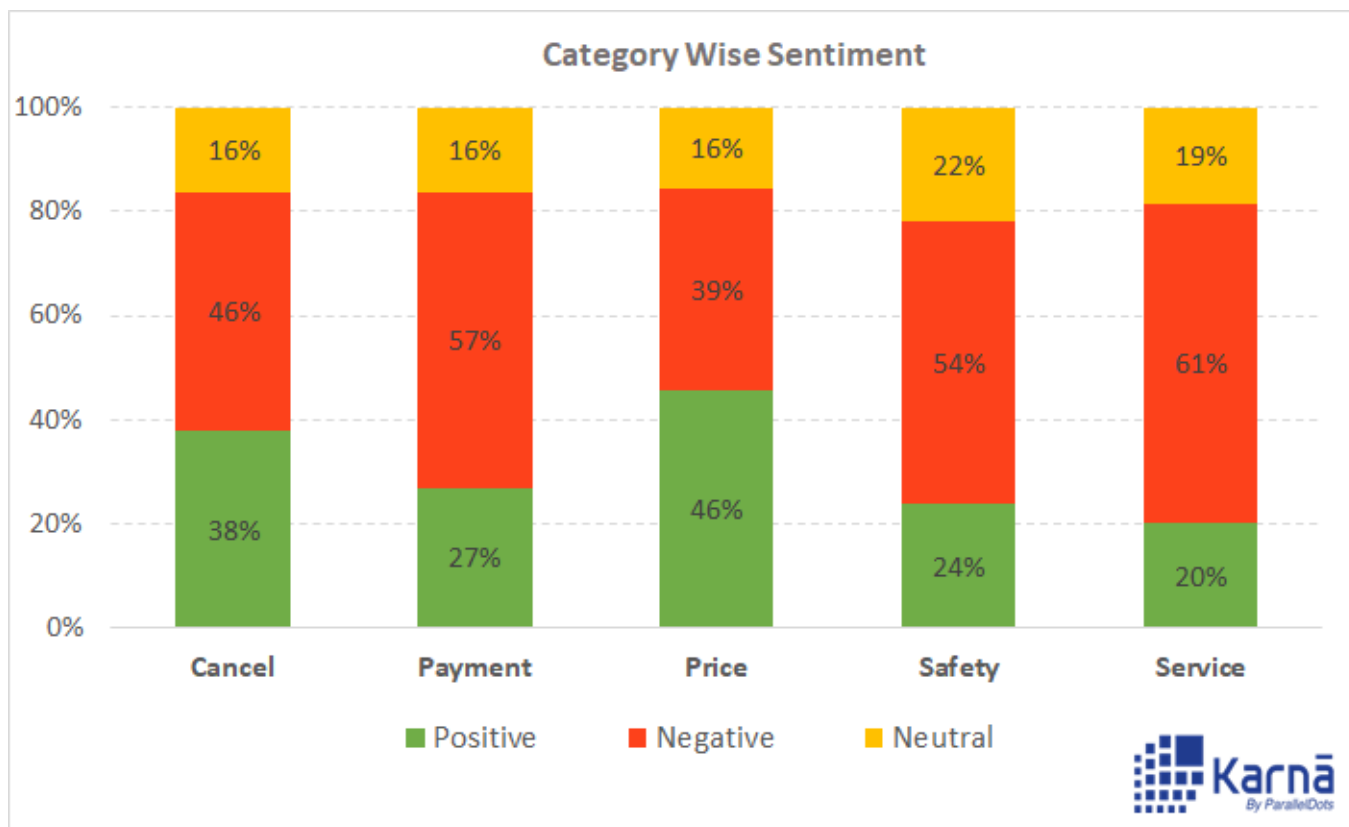


Рис. 1.11. Аналіз коментарів про сервіси компанії Uber

На цьому рисунку можна побачити порівняльну характеристику коментарів відносно різних сервісів компанії Uber в більшості соціальних мереж.

Facebook – американська онлайн-соціальна мережа та сервіс соціальних мереж, що належить Meta Platforms. Заснована в 2004 році Марком Цукербергом й іншими випускниками Гарвардського університету. На момент 2020 року в мережі зареєстровано більш ніж 2.8 мільярди активних користувачів. Соціальна платформа, де коментарі пишуть різні люди на будь-яку тему, новин, рекомендацій, реклами та будь-якого іншого типу спаму. Аналіз контенту на платформі можна побачити на рис. 1.12.

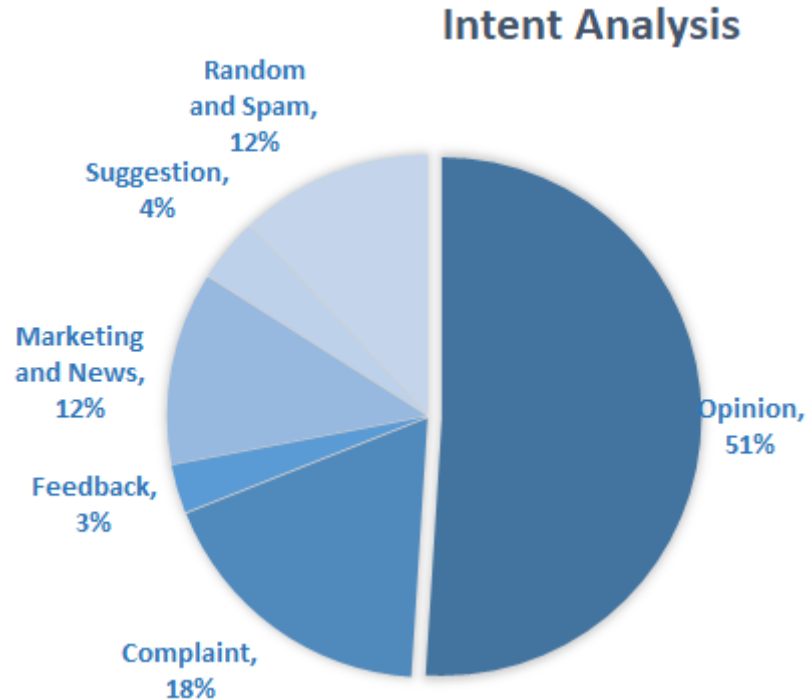


Рис. 1.12. Аналіз коментарів Facebook

На цьому рисунку можна побачити порівняльну характеристику коментарів на платформі Facebook.

З стрімким розвитком технологій настала епоха отримання значущих уявлень із даних соціальних мереж. Приклад аналізу коментарів про компанію Uber показав наскільки семантичний пошук є потужним. З цього можна зробити висновок, що компаніям бажано почати використовувати дані алгоритми й аналізувати коментарі користувачів на різних платформах та соціальних мережах.

1.4 Огляд недоліків та можливі їх покращення

В наш час є така проблема як велика кількість інформації. В даному випадку більше не означає краще. У компаній є значна кількість користувацький коментарів які необхідно зберігати й якось обробляти. В той же час для звичайних людей неможливо обробляти таку велику кількість даних й повідомлень. Через це більшість

компаній дуже часто не приймають необхідний рішень, які б допомогли їх зберегти репутації та покращити якість існуючих сервісів.

В той час як алгоритми аналізу визначають більшість коментарів коректно, все ж у них є недоліки. Комп'ютерні програми мають свою проблеми з розпізнаванням сарказму, іронії, заперечень, жартів та перебільшень. Всі попередньо перелічені пункти є проблемними для алгоритмів, адже вони потребують специфічної обробки, для отримання правильного результату.

Отримання некоректних результатів можна побачити на наступному прикладі. Слово 'засмучений' може бути класифікованим як негативне по наступним пунктам. Коли користувач використовує його в наступному реченні: “Я не засмучений”, то воно повинне бути класифікованим як позитивне.

Це повинно бути легким для визначення сарказму в реченнях як: “Мені подобається великий басейн у моєму готелі”, якщо це повідомлення супроводжується фотографією невеликого басейну; в той же час системи аналізу тексту не зможуть зрозуміти даного контексту, адже враховують тільки текст й не беруть до уваги картинки, як результат визначать повідомлення позитивним.

Текст без “шуму”.

Якщо загалом брати текст, то більшість з нього, такий як пишуть в соціальних мережах Twitter й іноді на Facebook може бути складним для аналізу, адже там не важко знайти будь-який контекст для якісного й коректного аналізу. Однак, взагалі, Twitter є гарною платформою й має задовільну репутацію для аналізу тексту, а зі збільшенням кількості слів у повідомленні стане кориснішим.

Одним з можливих способів покращення роботи систем, є видалення усього зайвого або ”шуму”. Для початку необхідно зрозуміти, що алгоритми не беруть до уваги абсолютно весь текст, а аналіз робиться тільки на основі частини з них. Прикладом зайвого можуть бути емодзі, адже вони вносять невизначеність при аналізі. Яким чином повинно бути визначено речення, яке користувач написав

негативним, але додав у кінці позитивний емодзі: чи це позитивне або негативне речення?

Усунення слів з надзвичайно низькою частотою.

Додатковим покращенням є видалення усіх слів, кількість яких є мінімальною, в порівнянні з іншими, адже вони не мають ніякого впливу при аналізі тексту. Видаливши такі слова, можна помітити, що результат класифікації став кращим, а продуктивність моделі збільшилась. Як приклад, проаналізувавши повідомлення на рисунку 1.14 можна побачити, що частота слів “fig” та “dale” є мінімальною. При цьому якщо ми виберемо пороговим значенням 10, всі слова з кількістю меншою будуть видалені.

| Term | Frequency | | Term | Frequency |
|---------|-----------|--|------------|-----------|
| played | 3422 | | dale | 2 |
| running | 455 | | fig | 2 |
| awesome | 417 | | <u>erw</u> | 1 |
| complex | 100 | | <u>tzx</u> | 1 |

Рис. 1.14. Приклад аналізу частоти використання слів

На цьому рисунку можна побачити, що слова такі як “dale”, “fig” та інші будуть видалені, адже їх частота менша за 10, яке є пороговим значенням.

Нормалізація слів.

Слова, які є похідними від інших повинні бути сконвертовані до базової форми, або ж інфінітиву. Цей підхід називається Лематизація. Як приклад, уявімо, що в тексті є наступні слова: Playing, player, plays, play, plays, played. Всі вони повинні бути нормалізовані до слова play, це зробить процес класифікації кращим а час роботи системи швидшим.

Використання складних ознак: n-грами та теги частини мови.

В деяких випадках, особливості як комбінація слів можуть внести більший вклад для класифікації, ніж у випадках, коли всі слова беруться до уваги незалежними одне від одного. Комбінації з N слів разом називаються N-грамами. Відомо, що біграми є найкращим прикладом N-грам. Приклад побудови таких N-грам можна побачити на рисунку 1.15.

| Old Features | Modified Features |
|--------------|-------------------|
| love | love phone |
| phone | ate apple |
| apple | apple + NOUN |
| book | book + VERB |

Рис. 1.15. Приклад побудови N-грам

На цьому рисунку можна побачити, що слово “book” використовується як іменник, але може бути дієсловом. У прикладі іменника може бути використано в реченні “book of N pages”, а як дієслово у “book a ticket”.

В кінці треба зазначити, що всі вищеперераховані методи використовуються для покращення існуючих систем для аналізу та класифікації тексту й допомагають вирішити існуючі проблеми, такі як: класифікація сарказму, визначення позитивного чи негативного відтінку та покращення швидкості роботи системи. Все це призводить до підвищення точності роботи на від 10% до 20%.

2 ТЕХНОЛОГІЇ ТА АЛГОРИТМИ КЛАСИФІКАЦІЇ ТЕКСТУ

Аналіз настроїв і думок набуває вирішальної ролі як у комерційних так і в дослідницьких програмах через їх можливу застосовність у кількох різних сферах. Тому велика кількість компаній включили аналіз думок і настроїв клієнтів як частину своєї місії. Одним з найцікавіших застосувань цих підходів є автоматичний аналіз повідомлень соціальних мереж на основі виражених почуттів та емоцій.

2.1 Нейронні мережі

Нейронні мережі також відомі як штучні нейронні мережі або ж змодельовані нейронні мережі. Вони є підмножиною машинного навчання і лежать в основі алгоритмів глибокого навчання. Їх назва та структура натхненні людським мозком, імітуючи спосіб, яким біологічні нейрони сигналізують один одному.

Штучні нейронні мережі складаються з шарів, які в свою чергу – з вузлів. Нейронні мережі містять вхідний шар, один або кілька прихованих і один вихідний. Кожен вузол або штучний нейрон з'єднується з іншим і має відповідну вагу та поріг. Якщо вихід будь-якого вузла перевищує вказане порогове значення, цей вузол буде активованим та надсилати дані на наступний рівень в мережі. В іншому випадку дані не передаються на наступний рівень мережі. На рисунку 2.1 можна побачити загальну схему нейронних мереж.

Нейронні мережі покладаються на навчальні дані для того, щоб з часом вивчати та покращувати свою точність. Проте, як тільки ці алгоритми навчання будуть точно налаштовані на точність, вони стануть потужними інструментами в інформатиці та штучному інтелекті, що дозволить нам класифікувати та кластеризувати дані з високою швидкістю. Завдання розпізнавання мовлення або зображення можуть займати хвилини та години, якщо порівнювати з ручною ідентифікацією експертів. Однією з найвідоміших нейронних мереж є пошуковий алгоритм Google.

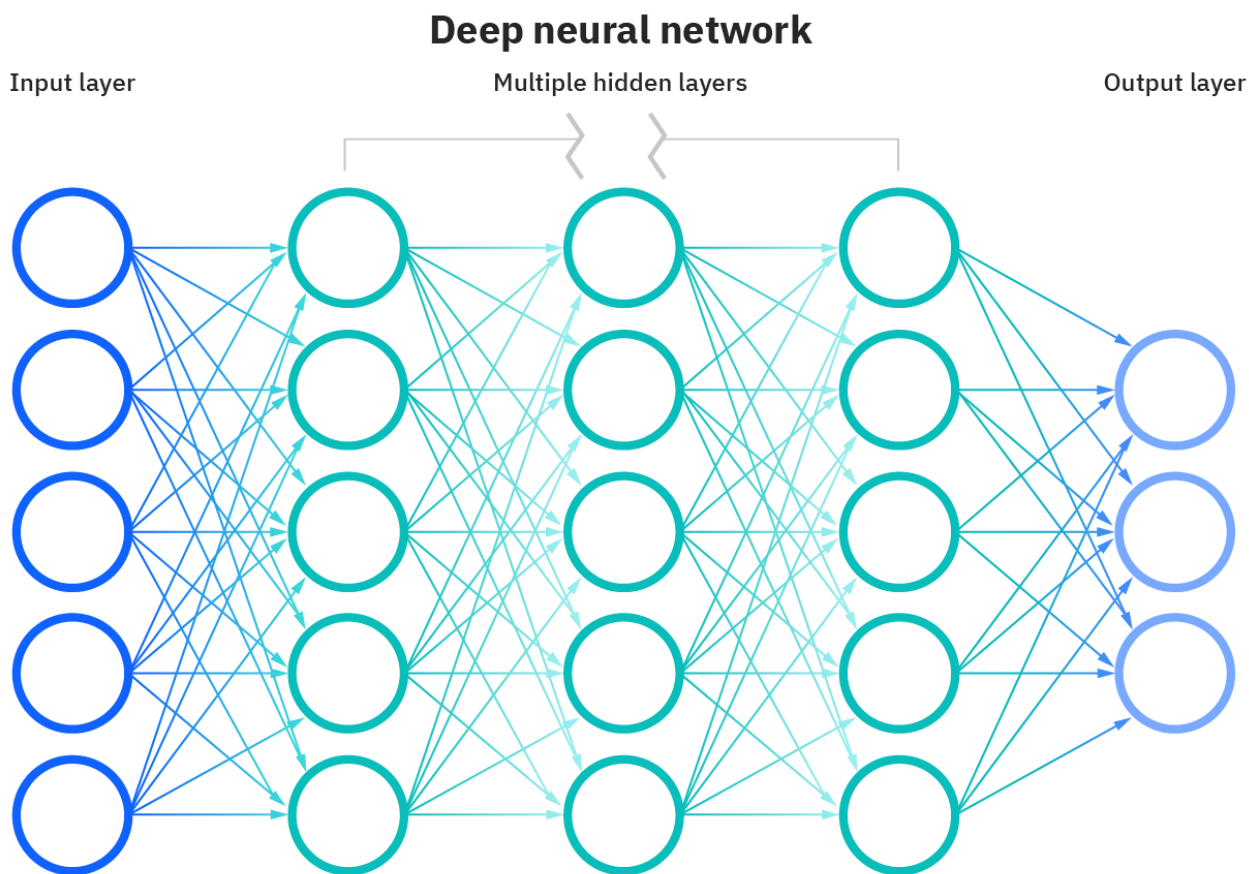


Рис. 2.1. Схема нейронної мережі

Якщо почати думати про кожний окремий вузол, як про власну модель лінійної регресії, що складається з вхідних даних, вагових коефіцієнтів, зміщення (або порогу) та вихідних даних, то розрахунок можна провести за наступною формулою.

$$\sum_{i=1}^m w_i x_i + bias = w_1 x_1 + w_2 x_2 + w_3 x_3 + bias$$

Далі можна побачити, як буде розраховано вихідне значення для кожного вузла.

$$\text{output} = f(x) = \begin{cases} 1 & \text{if } \sum w_1 x_1 + b \geq 0 \\ 0 & \text{if } \sum w_1 x_1 + b < 0 \end{cases}$$

Після визначення значень вхідного шару призначаються ваги. Ці коефіцієнти допомагають визначити важливість будь-якої змінної, причому більші з них роблять більш значний внесок у результат порівняно з іншими вхідними параметрами. Потім усі вхідні дані помножуються на відповідні ваги, а потім підсумовуються. Після цього вихідні дані передаються через функцію активації, яка визначає вихід. Якщо цей вихід перевищує заданий поріг, він “запускає” (або активує) вузол, передаючи дані наступному шару в мережі. Це призводить до того, що вихід одного вузла стає на вхід наступного вузла. Цей процес передачі даних від одного шару до наступного визначає цю нейронну мережу як мережу прямого зв’язку.

Для прикладу можна застосувати дану концепцію до реального прикладу, такого як: чи варто займатись серфінгом? Можливими результатами будуть так – 1 та ні – 0. На прийняття рішення будуть впливати три фактори: чи підходящі хвилі для серфінгу (x_1), чи багато людей (x_2) та чи були нещодавно напади акул (x_3). Кожне з цих значень буде вхідним, та наприклад, візьмемо $x_1=1$, $x_2=0$, $x_3=1$. Тепер необхідно присвоїти якісь значення для ваг для того, щоб визначити важливість кожного з параметрів. Чим більше значення ваги, тим значущим параметр є під час прийняття рішення. Уявімо, що результати наступні $w_1=5$, $w_2=2$, $w_3=4$. Тепер необхідно визначитись зі пороговим значенням. Нехай воно буде дорівнювати 3, й визначить, що значення зміщення буде дорівнювати -3. Отже, застосовуючи формулу на рис. 2.1 буде отримано значення 6. Також, звернувши увагу на формулу на рис. 2.2 й використавши дану функцію активації, можна буде визначати, що вихідним результатом цього вузла буде 1, адже 6 більше за 0. З цього можна зробити висновок, що нам варто займатись серфінгом, але якщо ми відрегулюємо ваги або поріг, ми зможемо досягти різних результатів від моделі. Коли ми спостерігаємо за

одним рішенням, як у прикладі наведеному вище, ми бачимо, як нейронна мережа може приймати все більш складні рішення залежно від результату попередніх рішень або шарів.

В вищезазначеному прикладі, використовувався перцептрон для того, щоб проілюструвати базові формули. В реальних прикладах використовуються сигмовидні нейрони, які відрізняються значеннями від 0 до 1. Оскільки нейронні мережі поведуться так само, як дерева рішень, передаючи дані з одного вузла до іншого, тож значення X від 0 до 1 зменшить вплив будь-якої даної зміни однієї змінної на вихід будь-якого вузла, а згодом і на вихід нейронної мережі.

Якщо почати думати про практичні приклади для нейронних мереж, таких як розпізнавання картинок або ж класифікація. Тож в таких задачах використовується навчання з учителем та набором даних, який містить вже результуюче значення. Під час навчання моделі ми будемо оцінювати її точність за допомогою функції витрат. Також це називається середньоквадратичною помилкою. Формула називається таким чином, адже це розраховується як квадрат різниці між значенням результуючим (з набору даних) та тим яке було отримано при розрахунках. Нижче можна побачити дане рівняння.

$$\text{Cost Function} = \text{MSE} = \frac{1}{2m} \sum_{i=1}^m (\hat{y} - y)^2$$

Зрештою, головною метою є мінімізація функції витрат для того, щоб забезпечити правильність відповіді для будь-якого даного спостереження. Коли модель коригує свої ваги та зміщення, вона використовує функцію вартості для навчання з підкріпленням, щоб досягти точки зближення або локального мінімуму. Процес у якому алгоритм коригує свої ваги називається градієнтним спуском й дозволяє моделі визначити напрямок в якому слід рухатись, щоб зменшити помилки (або ж мінімізувати функцію вартості). З кожним навчальним прикладом параметри

моделі коригуються, щоб поступово зближатися до мінімуму. На рисунку 2.2 зображено процес коригування параметрів.

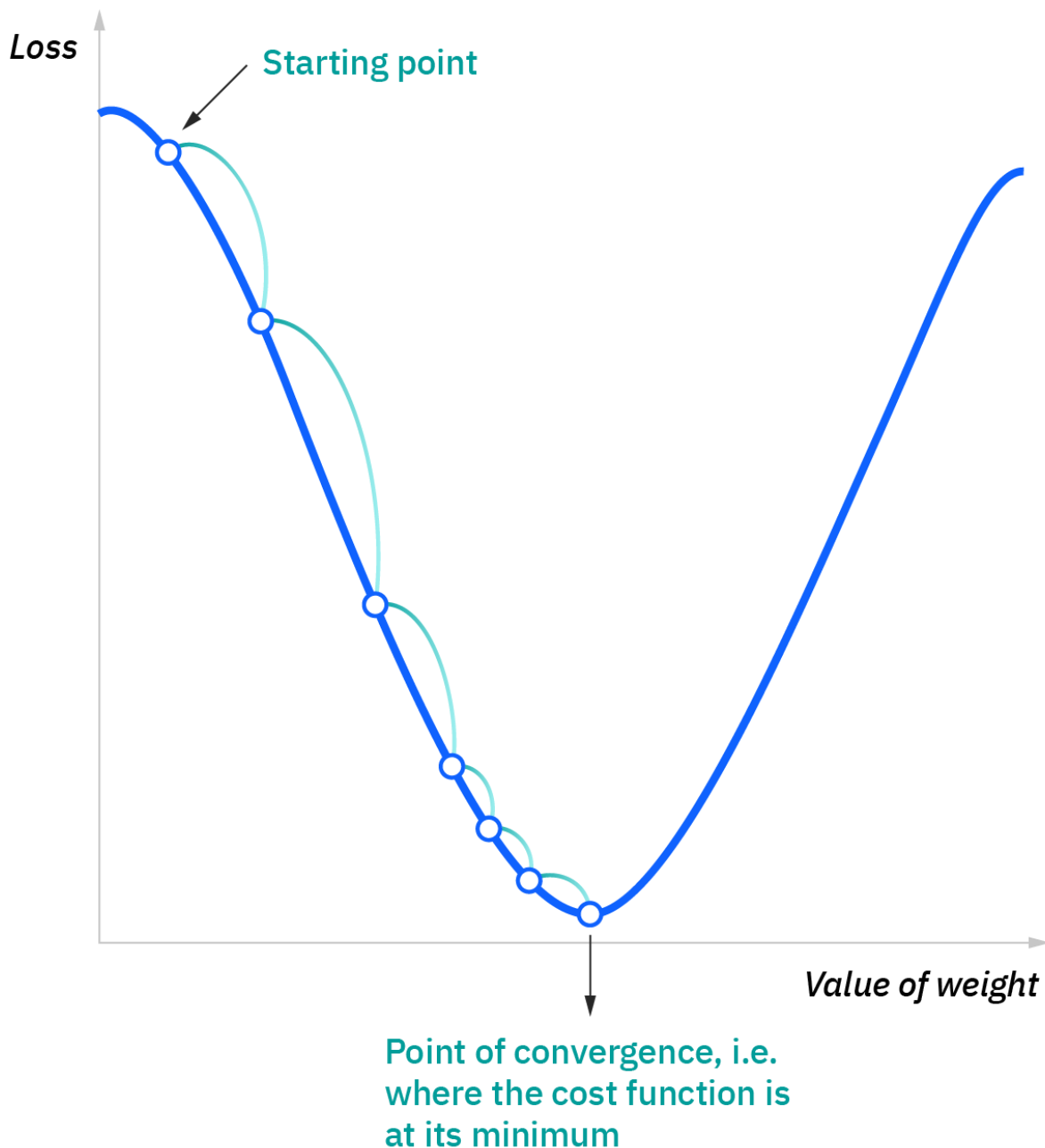


Рис. 2.2. Процес коригування значень параметрів

На рисунку зображено процес коригування параметрів, або ж називається процесом градієнтного спуску.

Більшість глибоких нейронних мереж мають прямий напрямок навчання, тобто вони протікають в одному напрямку: від входу до виходу. Однак можливо навчати модель шляхом зворотного поширення, тобто рухатись в протилежному напрямку: від виходу до входу. Зворотне розповсюдження дозволяє розрахувати значення помилки для кожного з нейронів й дозволяє належним чином налаштувати параметри моделі.

Існують різні типи нейронних мереж, які використовуються для різних цілей. Найбільш популярними є одношарові та багатшарові перцептрони, згорткові мережі (CNN) та повторювані нейронні мережі (RNN). Найпершим було створено перцептрон, який можна побачити на рисунку 2.3.

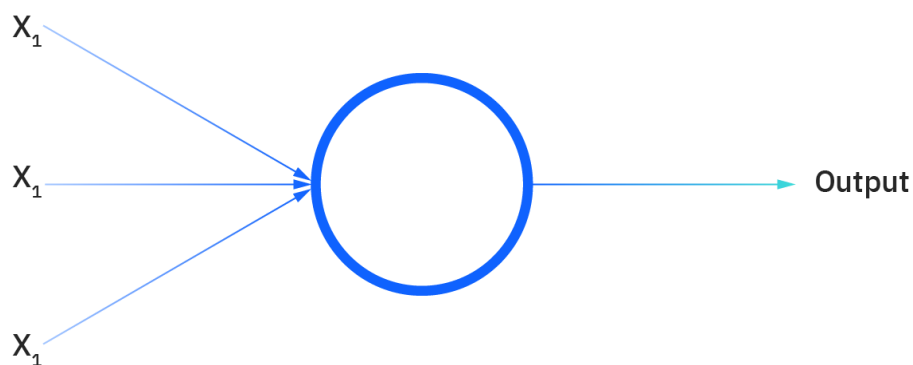


Рис. 2.3. Одношаровий перцептрон

Мережі прямого зв'язку або багатшарові перцептрони складаються з вхідного, прихованого та вихідного шарів. Хоча ці нейронні мережі називають багатшаровими перцептронами, важливо зазначити, що вони складаються з сигмовидних нейронів, а не з перцептронів, оскільки більшість проблем реального світу є нелінійними. Дані зазвичай подаються в ці моделі для їх навчання й вони є основою для комп'ютерного зору, обробки природної мови та інших нейронних мереж.

Згорткові нейронні мережі (CNN) подібні до мереж з прямим зв'язком, але вони

завичай використовуються для розпізнавання зображень, візерунків та комп'ютерного зору. Ці мережі використовують принципи лінійної алгебри, зокрема множення матриці для визначення шаблонів у зображенні.

Рекурентні нейронні мережі (RNN) ідентифікуються за їх петлями зворотного зв'язку. Ці алгоритми навчання в першу чергу використовуються під час використання даних часових рядів для прогнозування майбутніх результатів, таких як прогнозування фондового ринку або прогнозування продажів.

2.2 Naive Bayes

Алгоритм класифікації Байєса являє собою алгоритм машинного навчання, який може бути використаний в широкому спектрі задач класифікації. Частіше за все застосунки включаються в себе фільтрування спаму, класифікацію документу, аналіз тексту та інші. Алгоритм базується на теорії Томаса Байєса. Даний тип класифікатора є дуже популярним, адже його легко програмно реалізувати за допомогою будь-якої мови програмування. Також його можна використовувати в застосунках, які потребують обробки даних в реальному часі через те, що всі розрахунки є швидкими. Зважаючи на це, він легко масштабується й використовується в застосунках з розрахунками в реальному часі, які потребують моментальної відповіді на запит користувача.

Метод класифікації називається наївним адже допускає, що особливості з яких складається модель є незалежними одна від одної. Це означає, коли ви змінюєте значення одного параметра то це не буде мати ніякого впливу на ймовірності інших особливостей, які використовуються в алгоритмі під час класифікації.

Популярним прикладом, який показує основи ймовірностей є задача з підкиданням монетки. Коли хтось підкидає монету то з однаковою ймовірністю вона може впасти й людина побачить орла або решку. Це означає, що ймовірність складає 50%. Схожою є задача, коли ви хочете розрахувати ймовірність того, що в результаті

підкидання кубика, у якого всього 6 граней, на верхній ви побачите 1. Очевидно, що ймовірність буде дорівнювати 0.166(6), адже для отримання результату необхідно розділити 1 на 6 граней. Отже, це були класичні приклади задач на класифікацію. Отже, коли ви кажете, що про умовну ймовірність події А згідно іншої події В, це означає те, що ймовірність А відбудеться вже після того як подія В завершиться. Цю ймовірність можна виразити за допомогою формули:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad (2.1)$$

де $P(B)$ – ймовірність виконання події В;

$P(A \cap B)$ – ймовірність виконання події А й В одночасно.

Головною метою цієї формули є розрахунок ймовірності виконання події А згідно події В.

Як приклад можна взяти задачу з викладачами та студентами. Вона й трохи складнішою за попередні. Уявімо, що є школа з 100 людьми. Вони можуть бути ‘Вчитель’ або ‘Студент’. Також вони можуть бути за статтю ‘Чоловік’ або ‘Жінка’. З наведеним нижче переліком 100 людей, необхідно знайти ймовірність того, що конкретна людина належить до категорії ‘Чоловік’ та ‘Студент’. На наведеному нижче рисунку 2.4 можна побачити розподіл людей в закладі.

| | Female | Male | Total |
|---------|--------|------|-------|
| Teacher | 8 | 12 | 20 |
| Student | 32 | 48 | 80 |
| Total | 40 | 60 | 100 |

Рис. 2.4. Розподіл людей в навчальному закладі

Для того, щоб розрахувати необхідну ймовірність ми можемо інтуїтивно відібрати тільки чоловіків, кількість яких складає 60 й після цього сконцентрувати увагу на тих 12, які є вчителями. Отже, для отримання необхідного результату, а саме ймовірності, що серед усіх людей в навчальному закладі навмання вибраний користувач буде чоловіком, а також вчителем. Для отримання результату необхідно поділити кількість чоловік, які є вчителями, а саме 12, на загальну кількість чоловіків в закладі, а саме 60. В результаті буде отримано 0.2.

Правило Байєса визначає, що коли ми знаємо ймовірність $P(X|Y)$, яку отримали з набору даних, то необхідно знайти ймовірність $P(Y|X)$. Для того, щоб зробити це, нам необхідно замінити A та B в формулі 2.1 з особливістю X та відповіддю Y . Для огляду, в тестовому наборі даних, є інформація про X , але про Y нічого не відомо. Для кожного рядка в наборі даних ви б хотіли розрахувати значення Y для події X , яка вже трапилась. На рисунку 2.5 можна побачити як працює правило Байєса.

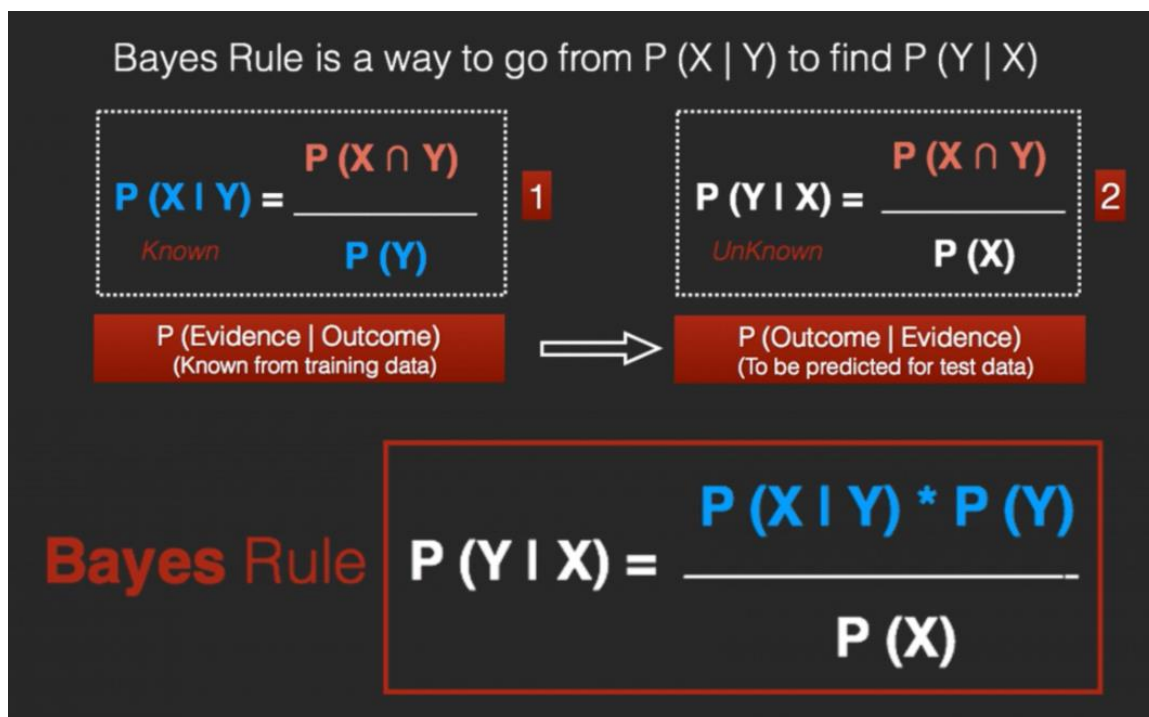


Рис. 2.5. Правило Байєса

Наївний алгоритм Байєса надає формулу для ймовірності параметру Y для конкретного параметру X . Коли параметри є незалежними, ми маємо змогу розширити правило Байєса. Алгоритм називається наївним адже головне допущення каже, що подія X є незалежною від будь-якої іншої. На рис. 2.6 можна побачити загальну формулу Naive Bayes.

The diagram consists of two main rectangular boxes on a dark background. The top box contains the mathematical formula for Naive Bayes:
$$P(Y=k | X_1..X_n) = \frac{P(X_1 | Y=k) * P(X_2 | Y=k) \dots * P(X_n | Y=k) * P(Y=k)}{P(X_1) * P(X_2) \dots * P(X_n)}$$
 The bottom box contains a conceptual breakdown:
$$\text{Probability of Outcome | Evidence (Posterior Probability)} = \frac{\text{Probability of Likelihood of evidence} * \text{Prior}}{\text{Probability of Evidence}}$$

Рис. 2.6. Формула Naive Bayes

На даному рисунку можна побачити загальну формулу алгоритму Naive Bayes з загальним описом, які параметри за що відповідають в формулі.

Вираз, який знаходиться зліва можна зрозуміти як ймовірність після виконання події. Перший доданок в чисельнику називається ‘Likelihood of Evidence’: описує ймовірність того, що параметр Y , кожного X , належить до певного класу C . Так як всі значення X вважаються незалежними одне від одного, ми перемножуємо їх. Останнім параметром в чисельнику є ‘Prior’, яке означає відношення кількості записів де значення параметру Y належить до класу C до загальної кількості записів. В знаменнику знаходиться параметр ‘Probability of Evidence’, який описує ймовірність виконання кожної події X незалежно одна від одної.

Уявімо, що є 1000 фруктів, які можуть бути одним із наступних: банан,

апельсин та інші. Все це є трьома можливими класами для параметру Y . Далі на рисунку 2.7 можна побачити загальний набір даних.

| Type | Long | Not Long | Sweet | Not Sweet | Yellow | Not Yellow | Total |
|--------|------|----------|-------|-----------|--------|------------|-------|
| Banana | 400 | 100 | 350 | 150 | 450 | 50 | 500 |
| Orange | 0 | 300 | 150 | 150 | 300 | 0 | 300 |
| Other | 100 | 100 | 150 | 50 | 50 | 150 | 200 |
| Total | 500 | 500 | 650 | 350 | 800 | 200 | 1000 |

Рис. 2.7. Загальний набір даних

На цьому рисунку можна побачити загальний набір даних з усіма категоріями.

Отже метою класифікатора є передбачення чи наданий фрукт є бананом, апельсином або ж іншим, коли відомо лише три параметри: довгий, солодкий та жовтого кольору. Головною метою є розрахунок трьох ймовірностей й та, яка буде відповідати конкретному фрукту. Фрукт, який буде мати найбільшу ймовірність буде переможцем.

Для початку необхідно розрахувати 'Prior' ймовірності для кожного з класів фруктів. Це є пропорція кількості кожного класу фруктів до загальної кількості фруктів. На прикладі банану можна зобразити сам алгоритм розрахунків. В наборі даних є 500 бананів й загальна кількість фруктів складає 1000. Тож для розрахунку значення необхідно розділити 500 на 1000 й в результаті отримати 0.5. Для апельсинів – 0.3, а для інших – 0.2.

Наступним кроком є розрахунок значення 'Probability of evidence', яке знаходиться в знаменнику. Це значення є добутком ймовірностей для всіх X . На прикладі параметру солодкості можна зобразити сам алгоритм розрахунків. В наборі даних є 650 солодких фруктів й загальна кількість фруктів складає 1000. Тож для розрахунку значення необхідно розділити 650 на 1000 й в результаті отримати 0.5.

Для довгих – 0.5, а для фруктів жовтого кольору – 0.8.

Передостаннім кроком є розрахунок параметру ‘Probability of likelihood evidences’. Цей значення є добутком ймовірностей трьох параметрів. На прикладі параметра довжини зобразимо роботу алгоритму. Всього в наборі є 500 бананів, а тих, що є довгими – 400. Для розрахунку ймовірності необхідно поділити 400 на 500 й отримати 0.8 для даного параметру. Аналогічно робиться для інших параметрів. Для солодких – 0.7, а для бананів жовтого кольору – 0.90. Після підраховується добуток усіх ймовірностей й отримується в результаті 0.504.

В результаті можна побачити загальний приклад, де вже підставили всі параметри в формулу. На рисунку 2.8 зображено приклад отримання результату.

Step 4: If a fruit is 'Long', 'Sweet' and 'Yellow', what fruit is it?

$$P(\text{Banana} | \text{Long, Sweet and Yellow}) = \frac{P(\text{Long} | \text{Banana}) * P(\text{Sweet} | \text{Banana}) * P(\text{Yellow} | \text{Banana}) * P(\text{banana})}{P(\text{Long}) * P(\text{Sweet}) * P(\text{Yellow})}$$

$$= \frac{0.8 * 0.7 * 0.9 * 0.5}{P(\text{Evidence})} = 0.252 / P(\text{Evidence})$$

$$P(\text{Orange} | \text{Long, Sweet and Yellow}) = 0, \text{ because } P(\text{Long} | \text{Orange}) = 0$$

$$P(\text{Other Fruit} | \text{Long, Sweet and Yellow}) = 0.01875 / P(\text{Evidence})$$

Answer: Banana - Since it has highest probability amongst the 3 classes

Рис. 2.8. Приклад Naive Bayes

На цьому рисунку можна побачити приклад визначення фрукту за допомогою Naive Bayes. Результатом є банан, адже він задовольняє всім параметрам: довжина, жовтий колір та солодкість. В прикладі не розраховувалось значення знаменника, адже він є однаковим для кожного класу й тож він не буде впливати на результат.

В вищенаведеному прикладі під час розрахунку значення $P(\text{Orange} | \text{Long, Sweet and Yellow})$ було отримано 0. Цей результат є логічним, адже в наборі даних

не було довгих апельсинів. Тож, коли ваша модель складається з великої кількості параметрів, загальна ймовірність буде дорівнювати 0 й все через один з параметрів. Для того щоб цього уникнути цього необхідно замінити в наборі даних всі значення, які дорівнюють 0 на яесь невелике, наприклад 0.1. Ця проста заміна значення матиме вплив на загальну ймовірність й вона не буде дорівнювати 0. Така заміна називається корекцією Лапласа. В багатьох реалізаціях алгоритму Naive Bayes, на різних мовах програмування, таке значення можливо вказувати.

2.3 Порівняльний аналіз вищеперерахованих методів

В минулих підрозділах були перераховані підходи та алгоритми для створення систем, які б могли класифікувати текст за настроєм як позитивний та негативний. У кожного є свої переваги та недоліки, які необхідно враховувати перед початком розробки програмного забезпечення.

Є три головних типи задач для класифікації тексту, а саме: класифікація за настроєм, класифікація за наміром та контекстний семантичний пошук. Так як в минулих розділах розглядалися методи, які стосуються класифікації за настроєм то краще буде розглядати їх плюси й недоліки.

Головними недоліками, які наявні в обох методах, стосуються розуміння емоційного забарвлення повідомлень, а саме: коли користувач використовує сарказм, іронію, заперечення, жарти та перебільшення. Нажаль, сарказм та іронію важко визначати, адже це є специфічними способами вираження емоцій, в більшості випадків які просто будуть розпізнані, як позитивні. В іншому ж випадку з запереченнями набагато легше, адже беручи до уваги специфіку різних мов, заперечувальні частки ставляться перед або після словами або ж дієсловами й для вирішення необхідно просто розширити аналіз від одного слова до двох.

Позитивним серед обох алгоритмів є те, що вони обидва підходять для застосунків, які потребують класифікації аналізу й класифікації у реальному часі.

3 РЕАЛІЗАЦІЯ СИСТЕМИ АНАЛІЗУ ТА КЛАСИФІКАЦІЇ ТЕКСТУ

Для реалізації системи роботи були обрані сучасні технології. Під час розробки було використано наступна платформа: .Net Core[8]. Для розробки системи рекомендацій було використано ML.Net. Весь код було написано за допомогою мови програмування C#[12-15].

Результуючий проект складається з декількох частин, а саме: Web API та клієнтського застосунку. Сам веб-сервер складається з контролерів, за допомогою яких клієнтський застосунок отримує данні. Також є дві робочі системи для класифікації тексту, перша використовує алгоритм naïve-bayes й друга використовує нейронну мережу. Клієнтський застосунок реалізовано за допомогою фреймворку Angular. Всі данні для навчання й тестування зберігались у різних txt-файлах. Всього в файлі налічується 10000 записів. Дані поділені на 80% для навчання й 20% для тестування. В процесі виконання застосунку й бібліотеки використовувалася система контролю Git.

3.1 Опис технологій

.Net Core — це платформа для розробки з відкритим кодом. Головною перевагою цієї платформи перед .Net Framework є те, що він є крос-платформеним. На цій платформі можна створювати застосунки для Windows, macOS та Linux. На рисунку 3.1 можна побачити логотип .Net Core.

Розробники цієї платформи намагаються максимально зберігати схожість з .Net Framework. Це все робиться для того, щоб існуючі застосунки було легше переносити з однієї платформи на іншу. Під час написання проєктів можна використовувати вже існуючий код, написаний іншими розробниками.



Рис. 3.1. Логотип .Net Core

Для цього лише потрібно додати посилання на необхідний NuGet-пакет у свій проект. Платформа .Net Core складається з таких частин:

- 1) .Net Runtime — система, що складається з Garbage Collector та системи за допомогою якої можна підключати інші пакети;
- 2) Fundamental Libraries — набір базових типів даних;
- 3) SDK & Compiler — набір компіляторів й середовища розробки за допомогою яких можливо створювати застосунки;
- 4) 'dotnet' app host — застосунок за допомогою якого можливо запускати інші застосунки написані на мові програмування C# або ж інших з сімейства .Net.

ML.Net — новий фреймворк для .Net розробників, який дає змогу використовувати машинне навчання у власних проектах. За допомогою цієї бібліотеки можна отримувати прогнози автоматично, на основі існуючих даних. ML.Net базується на моделі машинного навчання. Ця модель визначає які кроки необхідно виконати для отримання прогнозів, на основі вхідних даних. Розробники можуть навчити користувацьку модель вказав необхідний алгоритм навчання, а також за допомогою імпорту вже навченої моделі TensorFlow. За допомогою цієї

бібліотеки можливо отримувати такі прогнози як: класифікація, регресія, визначення аномалій, рекомендації, часові ряди та класифікація зображень.

C# — сучасна об'єктно-орієнтована мова програмування, яка є строго типізованою. В наш час вона підтримує можливість розробки застосунків, що базуються на компонентах (пакетах). Кожен з таких компонентів буде реалізовувати унікальний функціонал. Застосунок, створений за допомогою цієї мови програмування, буде надійним й стійким. За допомогою GC (Garbage Collector), зайві об'єкти будуть видаленими й застосунок не буде займати зайву пам'ять. За допомогою обробників виключень можна обробляти критичні ситуації. Також ця мова програмування є безпечною за допомогою строгої типізації, яка не дозволить те, що зможе призвести до аварійного завершення роботи застосунку. На рисунку 3.2 можна побачити логотип мови програмування C#.

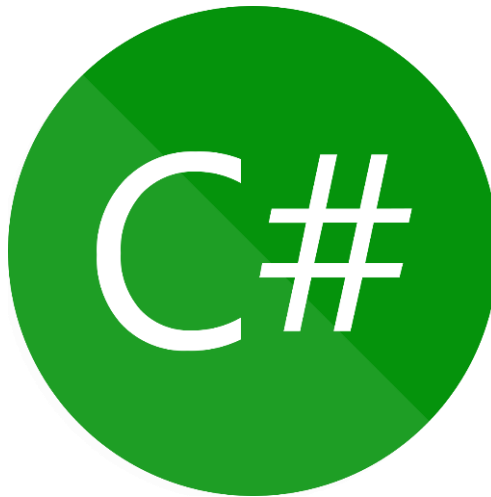


Рис. 3.2. Логотип C#

Angular — це платформа для розробки, яка побудована на базі мови програмування TypeScript. Як платформа Angular включає наступне:

- 1) фреймворк, який побудовано на компонентах для створення масштабованих веб-застосунків;
- 2) колекція інтегрованих бібліотек, які охоплюють широкий спектр функцій,

включаючи маршрутизацію, керування формами, зв'язок клієнт-сервер тощо;

3) набір інструментів для розробників, які допоможуть вам створювати, тестувати та оновлювати код.

На рисунку 3.3 можна побачити логотип Angular.



Рис. 3.3. Логотип Angular

TypeScript — це мова програмування розроблена та підтримувана компанією Microsoft. Це суворий синтаксичний набір над мовою JavaScript, який додає до мови обов'язкову статичну типізацію. TypeScript створено для розробки великих застосунків, які компілюються в JavaScript. Оскільки мова є надмножиною JavaScript, то вже існуючі програми також є дійсними програмами на TypeScript. Можна використовувати для розробки програм для виконання як на стороні клієнту (в браузері) так і на стороні серверу (як Node.js або Deno). На рисунку 3.4 зображено логотип TypeScript.



Рис. 3.4. Логотип TypeScript

Всі дані зберігаються в csv-файлах. Це особливий формат файлів, де данні розділяються спеціальним символами.

Git — це система контролю версій, за допомогою якої можливо швидко й ефективно писати код. Ще однією перевагою цієї системи є те, що за її допомогою декілька програмістів можуть працювати одночасно над проектом і не заважати один одному. На рисунку 3.5. можна побачити логотип Git.



Рис. 3.5. Логотип Git

Так як було зазначено, що .Net Core й Angular є крос-платформеним, то важливим кроком після реалізації застосунку є тестування того, що він буде працювати на інших платформах, крім операційної системи Windows. Для

перевірки цього функціоналу було вирішено проводити тестування на одному з Linux дистрибутивів.

Linux — це є одним з найпопулярніших прикладів програмного забезпечення, яке є вільним, тобто кожен може подивитись код системи та будь-яких інших її компонентів. Великою перевагою, в порівнянні з іншими операційними системами є те, що ці дистрибутиви є безкоштовними. Найчастіше операційна система використовується в виді серверної. Як приклад, 8 із 10 найпопулярніших компаній, що замаються веб-хостингом, використовують Linux. Логотип Linux можна побачити на рисунку 3.6.



Рис. 3.6. Логотип Linux

Під час тестування використовувався конкретний Linux дистрибутив, а саме Kubuntu 20.04.

Kubuntu — операційна система, в розробці якої приймають участь розробники з усього світу. Ця система є альтернативою до таких популярних як Windows та MacOS та містить усе необхідне для того, щоб було можливим працювати або ж

грати. Перевагою Kubuntu є те, що розробники поєднали Ubuntu з KDE та Plasma desktop. В результаті чого, після установки у користувача будуть усі необхідні застосунки для початку комфортної роботи. Логотип Kubuntu можна побачити на рисунку 3.7.

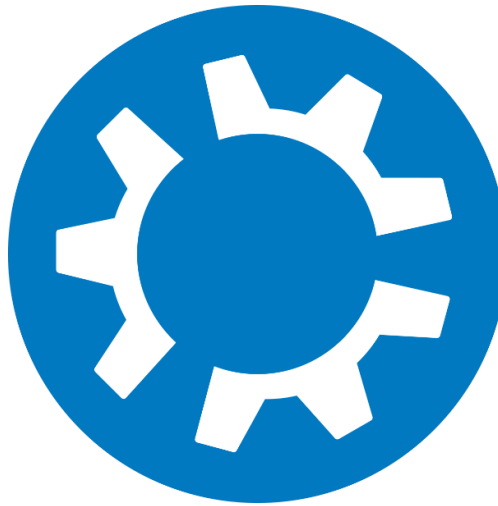


Рис. 3.7. Логотип Kubuntu

3.2 Опис Web API

Під час розробки було вирішено розділити його на декілька частин. Перша частина буде веб-клієнтом за допомогою якого у користувача буде можливість переглядати результати, то порівнювати алгоритми. Наступними частинами є плагіни, які реалізують алгоритми роботи з текстом, а саме: Naive Bayes та нейрона мережа. Останньою частиною буде Web API, який буде обробляти запити з клієнту та відправляти результати.

Веб-сервер складається з клієнтського застосунку, який використовується як SPA сервіс, контролерів та сервісів, які використовуються функціонал плагінів. До проекту були додані посилання на плагіни з реалізованими алгоритмами. Структуру проекту можна побачити на рисунку 3.8.

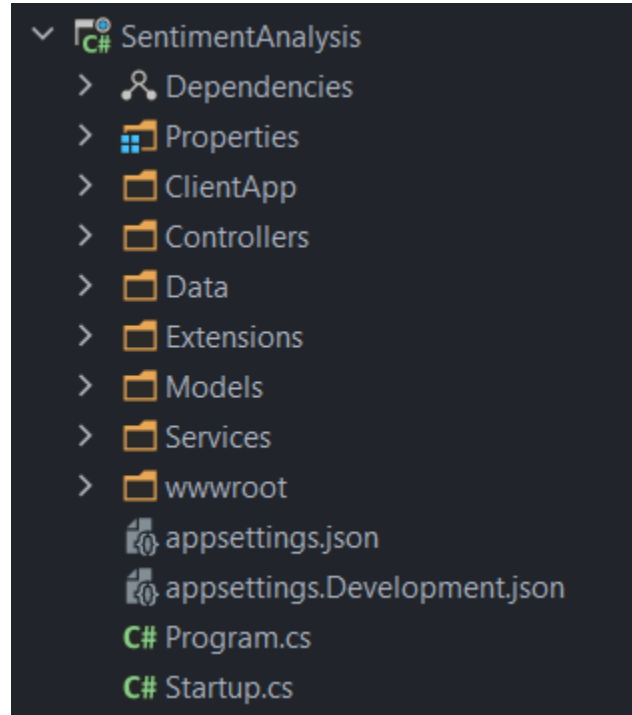


Рис. 3.8. Структура проекту веб-серверу

На цьому рисунку можна побачити структуру проекту. В директорії ClientApp знаходиться клієнтський веб- застосунок. Далі знаходиться директорія Controllers, в якій знаходяться всі контролери, які використовуються для обробки запитів, які надсилатиме клієнтський застосунок. В папці Services знаходяться усі сервіси для обробки даних, а також для виклику методів, які знаходяться в плагінах, й обробки результатів. В папці Extensions знаходяться методи для обробки результатів з сервісів й конвертація у моделі (директорія Models), які будуть зрозумілі на клієнтській частині.

Даний застосунок, окрім ASP.Net Core фреймворку використовує патерн MVC (Model-View-Controller). Цей патерн створений для того, щоб розділяти бізнес логіку застосунку з представленням користувача. Традиційно використовуваний для настільних графічних інтерфейсів користувача (GUI), цей шаблон став популярним для розробки веб-застосунків. Популярні мови програмування мають фреймворки MVC, які полегшують реалізацію шаблону.

3.3 Опис веб-застосунку

Для розробки клієнтського веб-застосунку було обрано один з найпопулярніших та потужніших фреймворків, а саме Angular. Перевагою цього фреймворку є те, що код буде працювати в усіх браузерах, отже й платформах. З цього можна зробити висновок, що більше користувачів матимуть змогу скористатись системою для аналізу й класифікації тексту. На рисунку 3.9 можна побачити структуру проекту.

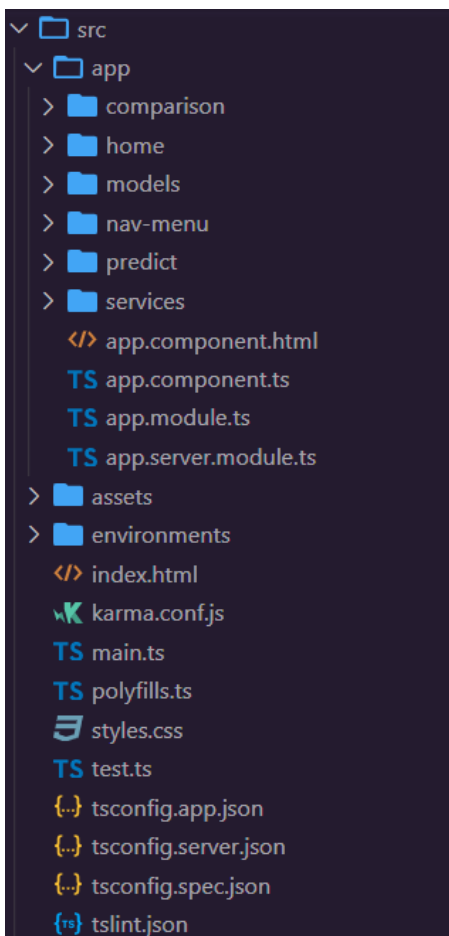


Рис. 3.9. Структура клієнтського веб-застосунку

На цьому рисунку можна побачити структуру проекту. Весь код застосунку знаходиться в директорії `app`, всі файли поза межами неї є конфігураційними. Всі компоненти знаходяться в окремих директоріях. Всього їх чотири, а саме:

comparison, home, nav-menu та predict. В директорії models не знаходяться інтерфейси, які є відображенням даних, отриманих й переданих до веб-серверу.

3.4 Опис плагіну нейронної мережі

Для зручної роботи з реалізованими алгоритмами було створено один інтерфейс, який містить один публічний метод Process. В якості вхідних параметрів метод приймає речення яке треба класифікувати, та додаткові налаштування для алгоритму. В якості результату повертається речення, його тип (позитивне або негативне), а також додаткові метрики, які були отримані під час виконання алгоритму та можуть бути цікаві користувачу.

Даний плагін використовує бібліотеку ML.NET з реалізованою нейронною мережею. Нажаль, дана бібліотека є новою й не занадто розвиненою в плані документації. В ній наявні реалізовані алгоритми та системи для різних типів повсякденних задач, але в той же час про їх реалізацію нічого не вказано. Єдине, що вдалося знайти про реалізацію це те, що для класифікації тексту використовуються нейронні мережі типу (RNN), а саме LSTM.

LSTM спеціально розроблені, щоб уникнути довгострокової проблеми залежності. Запам'ятовування інформації протягом тривалого періоду часу – це практично їхня поведінка за замовчуванням, а не те, що їм важко вивчити. З цього виходить, що дані мережі влучні під час розробки системи для класифікації текстів, адже речення за довжиною можуть бути, як короткими так і занадто довгими.

На рисунку 3.10 зображено код головного методу плагіну.

```

public PredictorOutput Process(PredictorInput input)
{
    _context = new();
    _fullData = LoadData(_context, input.Settings);
    _model = BuildAndTrainModel(_context, _fullData.TrainSet, input.Settings);
    _metrics = Evaluate(_context, _model, _fullData.TestSet, input.Settings);

    var prediction = MakePrediction(_context, _model, input);

    var messageType = prediction.Prediction ? MessageType.Positive : MessageType.Negative;

    return new()
    {
        MessageType = messageType,
        SentimentText = input.PredictionText,
        Probability = messageType == MessageType.Positive ? prediction.Probability : 1 - prediction.Probability,
        Metrics = new()
        {
            ["Accuracy"] = $"{_metrics.Accuracy:P2}",
            ["AreaUnderRocCurve"] = $"{_metrics.AreaUnderRocCurve:P2}",
            ["F1Score"] = $"{_metrics.F1Score:P2}"
        }
    };
}

```

Рис. 3.10. Код методу Process в плагіні

На цьому рисунку можна побачити метод обробки класифікації тексту за допомоги нейронної мережі. В самому початку створюється контекст мережі для обробки запиту. Після цього виконується завантаження даних з csv-файлу, які будуть поділені на 20% тестових та 80% тренувальних. В результаті обробки тексту отримується результат у відсотках, чи є він позитивним або негативним, а також додаткові метрики, які користувачу можуть бути цікавими.

3.5 Опис плагіну Naive-bayes

Аналогічно до плагіну нейронної мережі було створено реалізацію з алгоритмом Naive Bayes. Даний алгоритм значно відрізняється від нейронної мережі. Під час прогнозування не використовуються тестові та тренувальні набори даних. Для надання результатів класифікатор використовує два словники, перший з гарними словами, а інший з поганими. На рисунку 3.11 зображено код головного методу плагіну.


```
0+2 usages
public PredictorOutput Process(PredictorInput input)
{
    _documents ??= ReadDocumentsFromFile( category: "Bad", fileName: "Bad.txt")
        .Union(ReadDocumentsFromFile( category: "Good", fileName: "Good.txt"))//IEnumerable<Document>
        .ToList();//List<Document>

    _classifier = new(_documents);

    var (goodProbability:double, badProbability:double) = Predict(input.PredictionText, _classifier);

    return new()
    {
        Metrics = new(),
        SentimentText = input.PredictionText,
        MessageType = goodProbability >= badProbability ? MessageType.Positive : MessageType.Negative,
        Probability = goodProbability >= badProbability ? goodProbability : badProbability
    };
}
```

Рис. 3.11. Код методу Process в плагіні

На даному рисунку можна побачити як створюється набір документів, які відносяться до обох категорій Good та Bad. Після цього створюється класифікатор та виконується аналіз вхідного речення від користувача. На виході алгоритм повертає дві ймовірності, а саме належність тексту до класу Good та Bad. В коді робиться перевірка на те, яка з них більша. В разі якщо ймовірність належності до класу Good буде більше то речення буде класифіковано як Positive, інакше Negative.

4 ОПИС РОЗРОБЛЕНОЇ СИСТЕМИ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

Результатом написання магістерської роботи став готовий проект, а саме система, яка може проаналізувати та класифікувати текст. Для зручності реалізацію головних частин системи було винесено в окремі плагіни. Це було зроблено для того, щоб систему було можливо використовувати в різних проектах. Для демонстрації працюючої системи було створено два проекти: Web-API та Angular застосунки.

4.1 Опис застосунку

Після того як було розроблено плагіни й ніяких помилок не було виявлено, було вирішено створити Angular- застосунок для зручного перегляду результатів. Схема проекту зображена на рисунку 4.1.

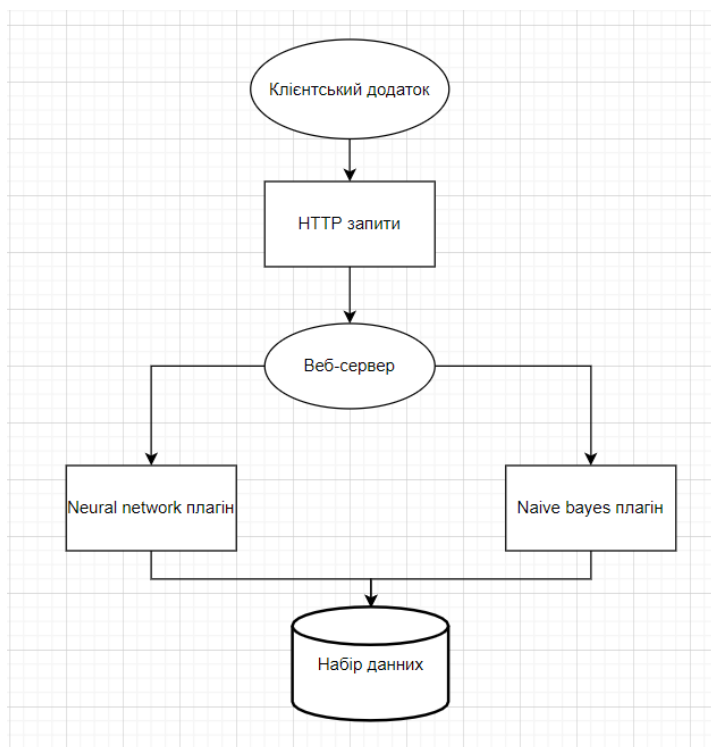


Рис. 4.1. Схема застосунку

На цьому рисунку можна побачити взаємодію між клієнтським застосунком і

веб-сервером.

Приклад даних, які використовувались для навчання й тестування системи можна побачити на рисунку 4.2.

```
Wow... Loved this place.    1
Crust is not good.         0
Not tasty and the texture was just nasty.    0
Stopped by during the late May bank holiday off Rick Steve recommendation and loved it. 1
The selection on the menu was great and so were the prices. 1
Now I am getting angry and I want my damn pho. 0
```

Рис. 4.2. Приклад набору даних

На цьому рисунку можна побачити набір даних, які використовувались для навчання й тестування системи.

При запуску застосунку на головній сторінці можна побачити перелік всіх алгоритмів (рисунок 4.3).

Sentiment Analysis Make prediction Algorithm comparison

Algorithms

Here is a list of sentiment analysis algorithms

| Algorithm Type | Description |
|----------------|---|
| Naive Bayes | Naive bayes based sentiment analyzer |
| Neural Network | Neural network based sentiment analyzer |

Рис. 4.3. Головна сторінка

На цьому рисунку можна побачити головну сторінку й перелік усіх алгоритмів й їх короткий опис.

Далі користувач може відкрити сторінку з порівнянням алгоритмів (рисунок 4.4). В таблиці є приклад речення й результати роботи двох алгоритмів, а

також оригінальний прогноз. Всі дані в даній таблиці є з базової вибірки, але які не використовувались для навчання або ж тестування. Оцінювання є різним для алгоритмів та базової вибірки. В базовій виборці, якщо оцінка ближча до одиниці – речення є негативним, та навпаки. В інших колонках зображено рівень впевненості який система отримала в результаті.

Algorithms comparison

Here is a comparison table for all algorithms based on data that hasn't been used during training

| # | Sentence | Expected | Naive Bayes | Neural Network |
|---|--|-----------------|-----------------|-----------------|
| 1 | I really like this product over the Motorola because it is allot clearer on the ear piece and the mic. | Negative (1.00) | Positive (1.00) | Negative (0.82) |
| 2 | However, the keypads are so tinny that I sometimes reach the wrong buttons. | Positive (0.00) | Positive (0.88) | Positive (0.64) |
| 3 | I exchanged the sony ericson z500a for this and I'm pretty happy with that decision. | Negative (1.00) | Positive (0.99) | Positive (0.98) |
| 4 | I was very impressed with the price of the cases. | Negative (1.00) | Positive (0.93) | |
| 5 | Also makes it easier to hold on to. | Negative (1.00) | Positive (0.88) | Negative (0.57) |

Рис. 4.4. Сторінка з порівняльною характеристикою алгоритмів

На цьому рисунку можна побачити сторінку з порівняльною характеристикою всіх алгоритмів що реалізовані в системі.

Останньою є сторінка для отримання передбачення (рисунок 4.5). Користувачу необхідно ввести речення й обрати алгоритм серед тих, які наявні в системі.

Make prediction

Select algorithm type and update default settings if needed to make prediction

Algorithm: Neural Network

Sentence: Good morning, sir!

Predict

Prediction response

Message type: **Positive**

Probability: **0.99**

Metrics list

Accuracy: **80.23%**

AreaUnderRocCurve: **88.43%**

F1Score: **81.27%**

Рис. 4.5. Сторінка з передбаченням

На цьому рисунку можна побачити сторінку з отриманням передбачення.

4.2 Опис діаграм

Для кращого розуміння структури проекту було вирішено створити IDEF3 діаграму, в якій буде описано усі процеси, наявні у застосунку. Перед початком створення діаграми треба зазначити, що IDEF3 є стандартом для документування технологічних процесів, як результат дослідження й моделювання сценаріїв є легшим. Послідовність етапів об'єкту, відносно процесу називається сценарієм. Виконання сценарію супроводжується двома документами. Першим описує деталі процесу, а саме структуру й послідовність. Другий описує послідовність виконання.

За допомогою цього стандарту можливо:

- 1) документувати наявні дані про технології процесу, які були отримані від тих, хто відповідає за процес;
- 2) аналіз впливу потоків документообороту на сценарій технологічного процесу;

3) допомога під час прийняття оптимальних рішень при організації технічних процесів.

Діаграми поділяються на два типи. Перший тип описує процеси, а саме послідовність етапів процесу. Інший тип описує стани й зміни об'єктів в процесі. Для кращого розуміння проекту було обрано перший тип діаграм. Перед початком побудови діаграми треба розібрати позначення. У таблиці 4.1 можна побачити усі позначення.

Таблиця 4.1

Позначення у стандарті IDEF3

| Позначення | Назва | Тип | Правила |
|------------|----------------|-------------|---------------------------------------|
| & | "І" | Розгортання | Дії в кінці обов'язково ініціюється |
| | | Згортання | Вихідні дії обов'язково завершуються |
| X | "Виключне АБО" | Розгортання | Лише одна дія в кінці ініціюється |
| | | Згортання | Лише одна вихідна дія завершується |
| O | "АБО" | Розгортання | Кінцеві дії можуть бути ініційованими |
| | | Згортання | Вихідні дії повинні бути завершеними |

На рисунку 4.6 можна побачити діаграму IDEF3, декомпозиції “Користувач”.

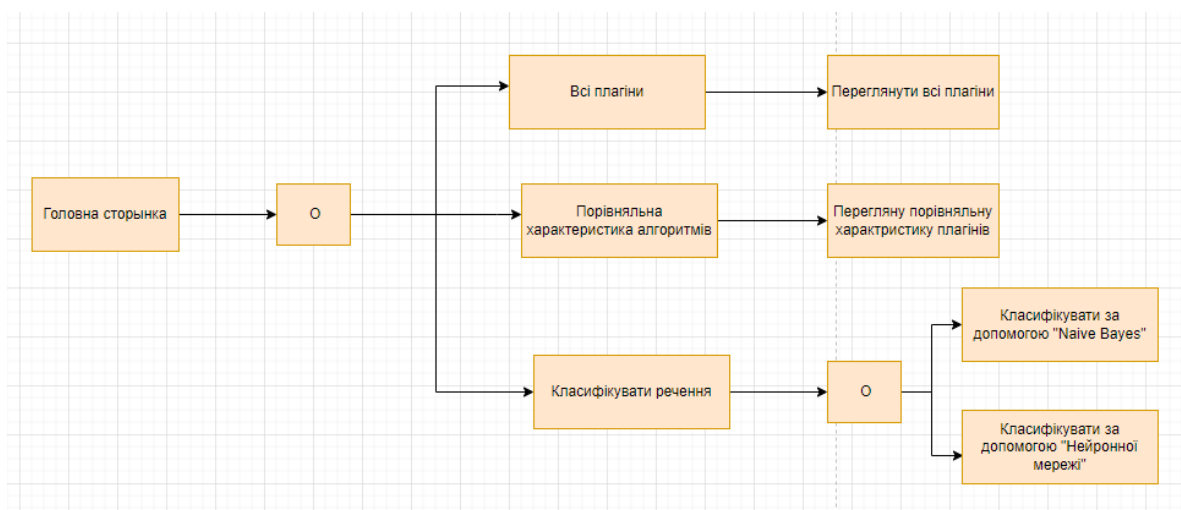


Рис. 4.6. Діаграма класів IDEF3, декомпозиція “Користувач”

На діаграмі зображено початкову дію, якою є “головна сторінка”. У цієї події є три можливих дії, які поєднані за допомогою “АБО”. Якщо користувач перейде на сторінку з рекомендованими фільмами, то у нього буде змога переглянути рекомендації. Якщо перейде до налаштувань, то буде можливість змінити налаштування системи рекомендацій. Якщо ж вибере сторінку з усіма фільмами, то матиме змогу переглянути усі фільми або ж переглянути деталі. При виборі деталей буде можливість переглянути статистику або ж подивитись на рейтинги фільму.

Ще однією важливою діаграмою, за допомогою новий користувач зміг би побачити, що можливо робити в застосунку є “Діаграма відносин між користувачем та прецедентом”. За допомогою цієї діаграми користувач може, не запускаючи застосунок. Після перегляду цієї діаграми користувач зможе вирішити, чи весь функціонал його задовольняє або ж є необхідність шукати інший застосунок, де буде більше можливостей. На рисунку 4.7 зображено діаграму відносин між користувачем й прецедентом.

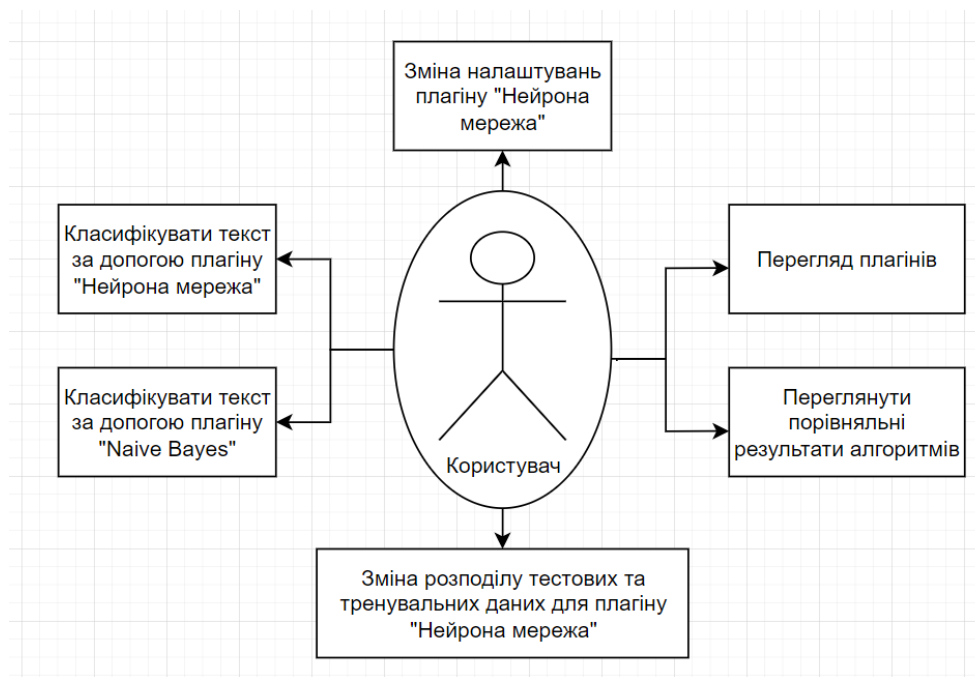


Рис. 4.7. Діаграма відносин між користувачем та прецедентом

На цьому рисунку можна побачити головні можливості, які є у користувача. Серед такого функціоналу у користувача є можливість переглянути всі плагіни, класифікувати текст за допомогою нейронної мережі, класифікувати текст за допомогою “Naive Bayes”. Також є можливість переглянути порівняльну характеристику результатів роботи двох алгоритмів.

ВИСНОВКИ

Під час написання магістерської роботи було проаналізовано існуючі методи, алгоритми й реалізації систем аналізу й класифікації тексту. Після чого було проведено порівняльну характеристику кожного й визначено їх переваги і недоліки.

В результаті виявлення недоліків, було запропоновано можливі рішення, які б могли покращити класифікацію. Одним з запропонованих методів покращення, що стосується аналізу, є використання декількох слів під час класифікації. Перевага даного рішення полягає в тому, що це допоможе вирішити проблему некоректної класифікації речень зі запереченнями. Наступним покращенням, що допоможе класифікувати текст ще краще є видалення усього зайвого “шуму” й слів які зустрічаються в реченні менш за все. За допомогою цього поліпшення якість аналізу може й трохи зміниться але час обробки буде значно покращено.

В магістерській роботі було реалізовано систему класифікації настроїв користувача по тексту. Для досягнення поставленої вирішено наступні завдання:

- 1) проаналізовано сучасний стан задачі аналізу тексту;
- 2) проведено огляд технологій та алгоритмів аналізу тексту, зокрема за настроєм;
- 3) розроблено систему класифікації настрою користувача з використанням обраних технологій;
- 4) проведено тестування розробленого застосунку.

Під час розробки проекту було детально описано головні складові системи класифікації. Перед початком реалізації було визначено технології й фреймворки, які будуть використовуватись в розробці плагінів й застосунків. Після визначення фреймворків їх було порівняно з існуючими аналогами й визначено недоліки й переваги. Було поетапно розписано реалізацію застосунку, починаючи з плагінів й закінчуючи готовим застосунком.

Результатом виконання магістерської роботи є повноцінна система класифікації й аналізу тексту користувачів. Реалізований застосунок включає в себе:

- 1) великий набір даних для навчання та тренування системи;
- 2) користувач-орієнтований інтерфейс, який є легким й зрозумілим навіть для тих хто вперше бачить цей застосунок;
- 3) систему класифікації й аналізу тексту, що містить весь необхідний для роботи функціонал;
- 4) реалізацію, що є швидкодіючою й клієнтський застосунок, який можна запуснути на будь-якій платформі;
- 5) тести, які перевіряють головний функціонал, який реалізовано у плагінах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Zhao Jianqiang, Gui Xiaolin, Zhang Xuejun. *Deep Convolution Neural Networks for Twitter Sentiment Analysis* 2018 URL: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8244338> (дата звернення: 24.11.2021).
2. Wei Xue, Tao Li. *Aspect Based Sentiment Analysis with Gated Convolutional Networks* 2018 URL: <https://arxiv.org/pdf/1805.07043.pdf> (дата звернення: 24.11.2021).
3. Li Yang, Ying Li, Jin Wang. *Sentiment Analysis for E-Commerce Product Reviews in Chinese Based on Sentiment Lexicon and Deep Learning* 2020 URL: <https://ieeexplore.ieee.org/abstract/document/8970492> (дата звернення: 24.11.2021).
4. Pollyanna Gonçalves, Matheus Araújo. *Comparing and Combining Sentiment Analysis Methods* 2014 URL: <https://arxiv.org/pdf/1406.0032.pdf> (дата звернення: 24.11.2021).
5. Yequan Wang, Aixin Sun. *Sentiment Analysis by Capsules* 2018 URL: <https://dl.acm.org/doi/pdf/10.1145/3178876.3186015> (дата звернення: 24.11.2021).
6. Alex Honchar. *Sentiment analysis: solutions and applications survey* 2017 URL: <https://hp-analytics.medium.com/sentiment-analysis-solutions-and-applications-survey-9e52d3ea2ac7> (дата звернення: 24.11.2021).
7. Savannah Krenzel. *Sentiment Analysis: A Definitive Guide* 2015 URL: <https://monkeylearn.com/text-classification/> (дата звернення: 24.11.2021).
8. База знань про .NET Framework. Документація по .NET: веб-сайт. URL: <https://docs.microsoft.com/> (дата звернення: 07.06.2020).
9. Shivam Bansal. *A Comprehensive Guide to Understand and Implement Text Classification in Python* 2018 URL: <https://www.analyticsvidhya.com/blog/2018/04/a-comprehensive-guide-to-understand-and-implement-text-classification-in-python/> (дата звернення: 24.11.2021).

10. Shashank Gupta. *Sentiment Analysis: Concept, Analysis and Applications* 2018 URL: <https://towardsdatascience.com/sentiment-analysis-concept-analysis-and-applications-6c94d6f58c17> (дата звернення: 24.11.2021).
11. Lexalytics. *Sentiment Analysis Explained*: веб-сайт. URL: <https://www.lexalytics.com/technology/sentiment-analysis> (дата звернення: 24.11.2021).
12. Відомості про використання .NET для створення застосунків на будь-якій платформі за допомогою C#. Документація по .NET: веб-сайт. URL: <https://docs.microsoft.com/> (дата звернення: 24.11.2021).
13. Ріхтер Дж. CLR via C#. Програмування на платформі Microsoft .NET Framework 4.5 мовою C#. СПб.: Питер, 2013. 896 с.
14. Албахарі Дж., Албахарі Б. LINQ. Кишеньковий довідник: навч. посіб. Петербург: БХВ-Петербург, 2009. 240 с.
15. Албахарі Дж., Албахарі Б. C# 8.0 Кишеньковий довідник: навч. посіб. Петербург: БХВ-Петербург, 2020. 240 с.
16. Jason Brownlee. *Deep Convolutional Neural Network for Sentiment Analysis (Text Classification)* 2017 URL: <https://machinelearningmastery.com/develop-word-embedding-model-predicting-movie-review-sentiment/> (дата звернення: 24.11.2021).
17. Ramesh Wadawadagi, Veerappa Pagi. *Sentiment analysis with deep neural networks: comparative study and performance assessment* 2020 URL: <https://link.springer.com/article/10.1007/s10462-020-09845-2> (дата звернення: 24.11.2021).
18. Sneh Paliwal, Sunil Kumar Khatri. *Sentiment Analysis and Prediction Using Neural Networks* 2018 URL: <https://ieeexplore.ieee.org/document/8597358> (дата звернення: 24.11.2021).
19. Olivier Habimana. *Sentiment analysis using deep learning approaches: an overview* 2020 URL: https://www.researchgate.net/publication/338351280_Sentiment_analysis_using_deep_learn

[ning approaches an overview](#) (дата звернення: 24.11.2021).

20. Brett Duncan. *Neural networks for sentiment analysis on Twitter* 2015 URL: https://www.researchgate.net/publication/308845323_Neural_networks_for_sentiment_analysis_on_Twitter (дата звернення: 24.11.2021).

21. Shiv Dhar. *Sentiment Analysis Using Neural Networks: A New Approach* 2018 URL: https://www.researchgate.net/publication/327936906_Sentiment_Analysis_Using_Neural_Networks_A_New_Approach (дата звернення: 24.11.2021).

22. Lilis Kurniasari. *Sentiment Analysis using Recurrent Neural Network* 2020 URL: https://www.researchgate.net/publication/339846409_Sentiment_Analysis_using_Recurrent_Neural_Network (дата звернення: 24.11.2021).

23. Alpna Patel. *Sentiment Analysis by using Recurrent Neural Network* 2019 URL: https://www.researchgate.net/publication/332435809_Sentiment_Analysis_by_using_Recurrent_Neural_Network (дата звернення: 24.11.2021).

24. Andrey Konstantinov. *Application of Neural Networks in Sentiment Analysis of Social Media Text Data* 2022 URL: https://www.researchgate.net/publication/357161433_Application_of_Neural_Networks_in_Sentiment_Analysis_of_Social_Media_Text_Data (дата звернення: 24.11.2021).

25. Rissa Shereen. *Sentiment Analysis Using Deep Learning* 2021 URL: https://www.researchgate.net/publication/350543179_Sentiment_Analysis_Using_Deep_Learning (дата звернення: 24.11.2021).

26. Sneha Paliwal. *Sentiment Analysis and Prediction Using Neural Networks* 2018 URL: https://www.researchgate.net/publication/331421691_Sentiment_Analysis_and_Prediction_Using_Neural_Networks (дата звернення: 24.11.2021).

27. V Umarani. *Sentiment Analysis using various Machine Learning and Deep Learning Techniques* 2021 URL: https://www.researchgate.net/publication/356624355_Sentiment_Analysis_using_various_Machine_Learning_and_Deep_Learning_Techniques (дата звернення: 24.11.2021).
28. Duane Szafron. *Explaining naive Bayes classifications* 2003 URL: https://www.researchgate.net/publication/228961211_Explaining_naive_Bayes_classifications (дата звернення: 24.11.2021).
29. Irina Rish. *An Empirical Study of the Naive Bayes Classifier* 2001 URL: https://www.researchgate.net/publication/228845263_An_Empirical_Study_of_the_Naive_Bayes_Classifier (дата звернення: 24.11.2021).
30. Trilochan Tripathy. *Bayes and Naive Bayes Classifier* 2014 URL: https://www.researchgate.net/publication/261368939_Bayes_and_Naive_Bayes_Classifier (дата звернення: 24.11.2021).
31. Paweł Cichosz. *Naive Bayes classifier* 2015 URL: https://www.researchgate.net/publication/313991281_Naive_Bayes_classifier (дата звернення: 24.11.2021).
32. Fitriana Harahap. *Implementation of Naive Bayes Classification Method for Predicting Purchase* 2018 URL: https://www.researchgate.net/publication/332074164_Implementation_of_Naive_Bayes_Classification_Method_for_Predicting_Purchase (дата звернення: 24.11.2021).
33. Державні санітарні правила і норми ДСанПіН 5.5.6.009-98 Влаштування і обладнання кабінетів комп'ютерної техніки в навчальних закладах та режим праці учнів на персональних комп'ютерах. Державні санітарні правила та норми(згідно з наказом від 30.12.1998 N 9).
34. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПіН 3.3.2.007-98(згідно з наказом від 10.12.98 № 7).

35. Наказ від 31.10.2016 № 287 Про затвердження ДБН В.1.1-7:2016 Пожежна безпека об'єктів будівництва. Загальні вимоги.
36. Катренко Л. А., Катренко А. В. Охорона праці в галузі комп'ютерингу. Львів: Магнолія-2006, 2012. 544 с.
37. Тепляков С. Патерни проектування на платформі .NET: навч. посіб. Питер: Питер 2020. 608 с.
38. Вайсфельд М. Об'єктно-орієнтоване мислення: навч. посіб. Питер: Питер 2018. 304 с.
39. Бхаргава А. Грокаем алгоритми. Ілюстроване посібник для програмістів і цікавих: навч. посіб. Питер: Питер 2019. 288 с.
40. Законодавство України про охорону праці //Збірник нормативних документів у 4 т. -К.: Основа, 2014 р.
41. ДСТУ 2293-99 Охорона праці. Терміни та визначення основних понять. Київ -1999 р.
42. Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, складності та напруженості трудового процесу, (Наказ МОЗ від 27.12.2006 р. № 528).
43. «Правила охорони праці під час експлуатації електронно-обчислювальних машин»(закон від 26.03.2010);
44. Гандзюк М. П., Желібо Е. П., Халімовський М. О. Основи охорони праці / За ред.. Гандзюка М. П. - К.: Каравела 2010 - 405 с.
45. Жидецький В.Ц. Основи охорои праці: Підручник. - К.: Основа, 2002. - 320 с.
46. Москальова В. М. Основи охорони праці. Підручник. - Київ: ВД Професіонал, 2005.-666 с.
47. Ткачук К. Н., Халімовський М. О., Зацарний В.В., та інші. Основи охорони праці: Підручник. -К.: Основа, 2013. -444 с.

48. Верников Г. Основи IDEF3. *Призначення IDEF3*. 2013. № 10. URL: <https://www.cfin.ru/vernikov/idef/idef3.shtml> (дата звернення: 24.11.2021).
49. Боровльова С. Ю., Дворецька С. В., Нездолій Ю. А., Цибенко Б. О. Мови об'єктно-орієнтованого програмування. Методичні вказівки до курсового проектування. Миколаїв: ЧДУ ім. Петра Могили, 2015. 12 с. URL: <http://lib.chdu.edu.ua/index.php?m=3&b=190> (дата звернення: 24.11.2021).
50. Нездолій Ю. О. Методичний посібник до курсових робіт з дисципліни «Технології програмування та створення програмних продуктів».
51. Державні санітарні правила і норми ДСанПіН 5.5.6.009-98 Влаштування і обладнання кабінетів комп'ютерної техніки в навчальних закладах та режим праці учнів на персональних комп'ютерах. Державні санітарні правила та норми (згідно з наказом від 30.12.1998 N 9).
52. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПіН 3.3.2.007-98 (згідно з наказом від 10.12.98 № 7).
53. Наказ від 31.10.2016 № 287 Про затвердження ДБН В.1.1-7:2016 Пожежна безпека об'єктів будівництва. Загальні вимоги.
54. Катренко Л. А., Катренко А. В. Охорона праці в галузі комп'ютерингу. Львів: Магнолія-2006, 2012. 544 с.