

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

Далекий Мирослав Русланович

«Алгоритм виявлення агресивної поведінки у повідомленнях на основі семантичного аналізу / An algorithm for detecting aggressive behavior during message exchange based on semantic text analysis»

спеціальність: 123 - Комп'ютерна інженерія
освітньо-професійна програма - Комп'ютерна інженерія
Кваліфікаційна робота

Виконав студент групи КІмз-21
М.Р. Далекий

Науковий керівник:
к.т.н., доц. О.Й. Піцун

Кваліфікаційну роботу допущено до захисту:

" ____ " _____ 20____ р.

Завідувач кафедри
_____Л. О. Дубчак

Тернопіль – 2023

РЕЗЮМЕ

Кваліфікаційна робота на тему “ Алгоритм виявлення агресивної поведінки у повідомленнях на основі семантичного аналізу ” зі спеціальності 123 «Комп’ютерна інженерія» освітнього ступеня «магістр» написана обсягом 83 сторінки та містить 28 ілюстрацій, 3 таблиці, 2 додатки та 50 джерел за переліком посилань.

Метою роботи є розробка алгоритму виявлення агресивної поведінки у повідомленнях на основі семантичного аналізу.

Методи досліджень. Для розв’язання поставлених задач у кваліфікаційній роботі використано методи: синтаксичного аналізу (для аналізу та опису текстових повідомлень); алгоритми обробки природніх мов (для виділення основних характеристик слів); об’єктно-орієнтованого програмування (для проектування програмного додатку оцінки рівня агресії в повідомленнях).

Результати дослідження: алгоритм визначення рівня агресії повідомлення на основі синтаксичного аналізу, програмний додаток аналізу та визначення рівня агресії текстового повідомлення на мові Python.

Результати роботи можуть бути використані при створенні систем моніторингу в соціальних мережах, системах обліку повідомленнями, для наукових досліджень та в навчальному процесі.

Орієнтовні напрямки розвитку досліджень: проектування алгоритмів автоматизованого визначення рівня агресії в голосових повідомленнях, створення нових програмних засобів та моделей для навчання.

КЛЮЧОВІ СЛОВА: АГРЕСІЯ, СОЦІАЛЬНІ МЕРЕЖІ, ОБРОБКА ПРИРОДНІХ МОВ, АНАЛІЗ ТЕКСТОВИХ ПОВІДОМЛЕНЬ.

RESUME

Graduate qualification work on A research management system in an automated microscopy system ” specialty 123 – Computer Engineering is 82 pages long and contains 28 illustrations, 3 tables, 2 appendices and 51 references.

The aim of the work is to develop an algorithm for detecting aggressive behavior in messages based on semantic analysis.

Research methods. To solve the tasks in the qualification work, the following methods were used: syntactic analysis (for analysis and description of text messages); natural language processing algorithms (for highlighting the main characteristics of words); object-oriented programming (for designing a software application for assessing the level of aggression in messages).

Research results: an algorithm for determining the level of aggression of a message based on syntactic analysis, a software application for analyzing and determining the level of aggression of a text message in Python.

The results of the work can be used in the creation of monitoring systems in social networks, messaging systems, for scientific research and in the educational process.

Approximate areas of research development: design of algorithms for automatic determination of the level of aggression in voice messages, creation of new software tools and models for training.

KEY WORDS: AGGRESSION, SOCIAL NETWORKS, NATURAL LANGUAGE PROCESSING, ANALYSIS OF TEXT MESSAGES.

ЗМІСТ

Вступ.....	7
1 Технологія семантичного аналізу в сучасному світі	10
1.1 Семантичний аналіз, підходи та сери застосування.....	10
1.2 Технологія машинного навчання	19
1.3 Програмні засоби для обміну та аналізу повідомлень	25
1.4 Постановка задач дослідження	28
1.5 Висновки до розділу	29
2 Методи та алгоритми визначення агресії в соціальних мережах.....	30
2.1 Алгоритми алгоритми обробки природніх мов	30
2.2 Алгоритми визначення агресивної поведінки в соціальних мережах .	39
2.3 Алгоритм визначення рівня агресії в текстових повідомленнях	45
2.4 Висновки до розділу	49
3 Програмний додаток оцінки рівня агресії в текстових повідомленнях	50
3.1 Структура програмного модуля аналізу текстових повідомлень	50
3.2 Функції обробки природніх мов.....	62
3.3 Тестування та аналіз реалізованої програми.....	66
3.4 Висновки до розділу	70
Висновки	71
Список використаної літератури	72
Додаток А Лістинг програмного програмного коду токнізації повідомлення ..	78
Додаток Б Світлокопії виданих публікацій.....	79
Додаток В Довідка про використання результатів кваліфікаційної роботи	83

ВСТУП

Актуальність роботи. Перехід людського спілкування на онлайн-платформи приносить багато переваг суспільству завдяки простоті публікації думок, обміну досвідом, миттєвому отриманню відгуків і можливості обговорювати найактуальніші теми. Крім того, це створює простір для антисоціальної поведінки, як-от переслідування, образа та мова ненависті. Виявлення такої антисоціальної поведінки є більш важливим для соціального добробуту через фінансові, законодавчі та соціальні причини. За щорічними даними Центру досліджень кібербулінгу в 2021 році, 33,8 % молодих людей у віці 12-17 років у США стикалися з кібербулінгом протягом свого життя. Це підкреслює важливість для онлайн-платформ пошуку рішення щодо того, як ідентифікувати АОВ. Ручне виявлення та моніторинг онлайн-контенту може бути дуже дорогим, тому нам потрібна автоматична процедура виявлення такої поведінки.

Термін АОВ – це загальний термін, який описує будь-яку зловмисну поведінку, яку можна знайти в текстовому вмісті на платформах онлайн-комунікації, як-от образи, погрози, особисті напади, використання шкідливої, грубої чи образливої лексики, кіберзалякування та зловживання. Ранні наукові дослідження щодо виявлення АОВ в онлайн-комунікації були здебільшого зосереджені на використанні традиційних методів машинного навчання (TML), таких як логістичні регресії, опорні векторні машини та дерева рішень, а також підходи на основі лексики. Ці методи значною мірою покладаються на широкую інженерію функцій, і продуктивність значною мірою залежить від представлення даних. Методи глибокого навчання (DL) автоматизують процедуру розробки функцій шляхом вивчення представлень даних за допомогою нелінійних перетворень. Такі представлення часто досягають набагато кращої продуктивності, ніж елементи ручної роботи. Тому задача розробки та

дослідження алгоритму виявлення агресивної поведінки у повідомленнях на основі семантичного аналізу є актуальною.

Метою роботи є розробка алгоритму виявлення агресивної поведінки у повідомленнях на основі семантичного аналізу.

Для досягнення даної мети ставились наступні завдання:

- провести дослідження технології семантичного аналізу;
- дослідити технології машинного навчання;
- провести аналітичний огляд існуючих програмних засобів обміну та передачі повідомлень;
- проаналізувати існуючі методи та алгоритми визначення агресії в повідомленнях;
- розробити алгоритм виявлення агресивної поведінки у повідомленнях на основі семантичного аналізу;
- реалізувати програмну систему автоматизованого аналізу повідомлень в соціальних мережах з метою виявлення агресивної поведінки

Об'єкт дослідження – процес аналізу природних мов.

Предмет дослідження – методи і алгоритми аналізу текстових повідомлень.

Наукова новизна одержаних результатів визначається наступним чином:

- проведено комплексний аналіз та класифікацію алгоритмів семантичного аналізу під час оцінки текстових повідомлень, що дало можливість виділити їх основні критерії при аналізі текстових повідомлень;
- розроблено алгоритм виявлення агресивної поведінки у повідомленнях на основі семантичного аналізу, що дало можливість на його основі спроектувати структуру програмного додатку аналізу текстових повідомлень на предмет виявлення агресивної поведінки.

Практична цінність одержаних результатів полягає в тому, що:

- розроблено структуру та здійснено дослідження структури програмної аналізу текстових повідомлень, що дозволило програмно реалізувати додаток для аналізу текстових повідомлень;

- реалізовано програмний додаток виявлення агресивної поведінки у повідомленнях на основі семантичного аналізу, що дозволило провести тестування запропонованого алгоритму та порівняти отримані результати з програмами-агалогами.

Публікації та апробація до випускної кваліфікаційної роботи. За результатами наукових досліджень, проведених у випускній кваліфікаційній роботі, підготовлено тези доповіді «Оцінка якості програмного коду на основі аналізу рівня його читабельності» обсягом 1 сторінка на VIII Науково-практичній конференції молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі», а також «Оцінка рівня агресії на основі аналізу текстових повідомлень» обсягом 1 сторінка на VIII Науково-практичній конференції молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі».

В першому розділі проведено дослідження особливостей технології семантичного аналізу та особливостей організації процесу машинного навчання.

В другому розділі було досліджено та описано сучасні методи та алгоритми виявлення агресії в текстових повідомленнях, а також запропонований алгоритм виявлення агресивної поведінки у повідомленнях на основі семантичного аналізу.

В третьому розділі наведено опис структури розробленого програмного модуля, проведено його моделювання та тестування.

1 ТЕХНОЛОГІЯ СЕМАНТИЧНОГО АНАЛІЗУ В СУЧАСНОМУ СВІТІ

1.1 Семантичний аналіз, підходи та сери застосування

Аналіз семантики – це процес витягування значення з тексту, що дозволяє комп'ютерам розуміти й тлумачити речення, абзаци чи весь документ. Цей процес включає в себе аналіз граматичної структури та встановлення зв'язків між словами в конкретному контексті. Семантичний аналіз є важливою частиною обробки природної мови (NLP) і є ключовим компонентом для різних інструментів машинного навчання, таких як чат-боти, пошукові системи та системи аналізу тексту. Інструменти, які використовують семантичний аналіз, можуть автоматизовано витягувати значущу інформацію з неструктурованих даних, таких як електронні листи, запити до служби підтримки та відгуки клієнтів. Семантичний аналіз враховує лексичну семантику, яка грає важливу роль у з'ясуванні взаємозв'язків між лексичними одиницями (рисунок 1.1).

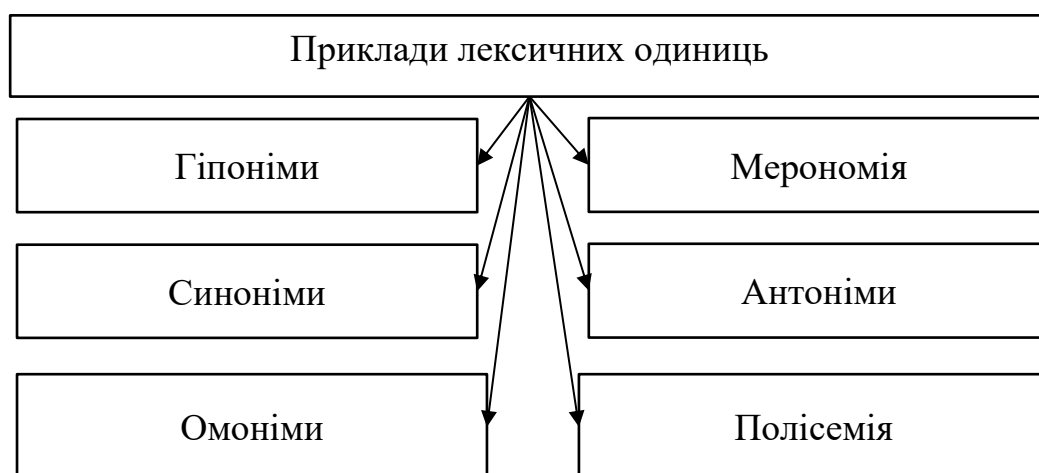


Рисунок 1.1 – Інформативні ознаки об'єктів на біомедичних зображеннях

Автоматизований семантичний аналіз використовує алгоритми машинного навчання. Шляхом навчання машин за допомогою семантично розширених алгоритмів на прикладах тексту можна досягти точних прогнозів на підставі

минулих спостережень. В процесі автоматизованого семантичного аналізу розв'язуються завдання, такі як розпізнавання сенсу слова, виправлення неоднозначностей та встановлення зв'язків в тексті. Для прикладу проаналізуємо слово «Апельсин». І можемо отримати три значення цього слова (рисунок 1.2): колір, фрукт, та цитрусове дерево.



а)



б)



в)

Рисунок 1.2 – Можливе значення слова «Апельсин» в залежності від контексту: колір (а), фрукт (б) та цитрусове дерево (в)

У контексті семантичного аналізу з використанням машинного навчання комп'ютери використовують усунення неоднозначностей слів для визначення правильного значення у конкретному контексті. Однією з важливих задач є вилучення зв'язків між різними елементами тексту.

Завдання вилучення зв'язків полягає в ідентифікації семантичних зв'язків, що присутні у тексті. Ці зв'язки зазвичай включають дві або більше сутності, такі як імена людей, місця, назви компаній тощо. Сутності пов'язані семантичною категорією, наприклад "працює в", "проживає", "є генеральним директором", "штаб-квартира". Наприклад, у фразі "Стів Джобс є одним із засновників Apple, штаб-квартира якої розташована в Каліфорнії" містяться два різних семантичних відносини.

*Стів Джобс (особа) є одним із засновників Apple (фірма)
штаб-квартира (фірма) якої розташована в Каліфорнії (місце)*

У семантичному аналізі з машинним навчанням використовуються різні моделі для досягнення конкретних цілей. Один із підходів - це моделі семантичної класифікації (рисунок 1.3), які включають:

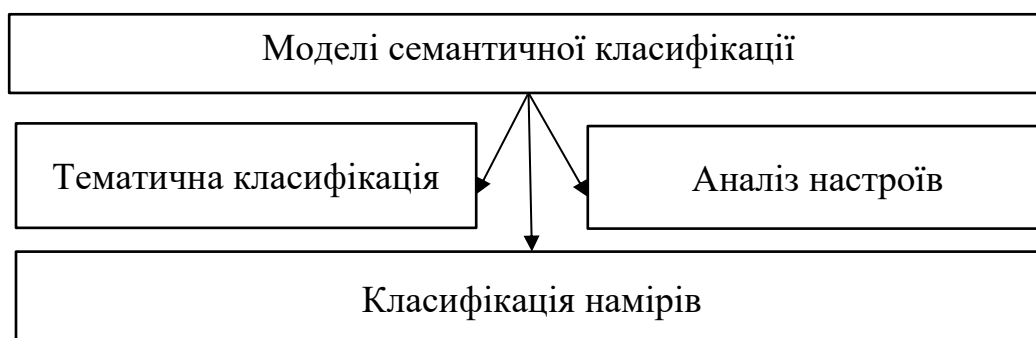


Рисунок 1.3 – Класифікація моделей семантичної класифікації

Модель тематичної сегментації передбачає сортування тексту за попередньо визначеними категоріями на основі його змісту. Наприклад, інструменти машинного навчання можуть класифікувати квитки служби підтримки за темами, такими як "Проблема з оплатою" або "Проблема з доставкою".

Модель "Аналіз настроїв" використовується для виявлення позитивних, негативних або нейтральних емоцій у тексті для визначення терміновості. Наприклад, можливо визначати настрої у Twitter для розуміння ставлення клієнтів до бренду.

На основі моделі класифікації намірів проводиться класифікація тексту на основі того, що клієнти хочуть зробити далі, наприклад, визначення електронних листів про продаж як "Зацікавлений" чи "Не зацікавлений".

Також використовуються моделі семантичного вилучення, такі як:

- Виділення ключових слів. Пошук відповідних слів і виразів у тексті, що надає детальну інформацію. Наприклад, можна аналізувати ключові слова у групі твітів, класифікованих як "негативні".

- Вилучення об'єктів. Ідентифікація іменованих об'єктів у тексті, таких як імена людей, компаній, місць і тощо.

Крім того, статистика, отримана на основі даних, допомагає виявляти сфери покращення та приймати кращі рішення, такі як створення міцної бази знань для розв'язання найпоширеніших проблем клієнтів.

В загальному дані які підлягають обробці можна класифікувати по мірі відповідності деякій структурі та принципам обробки. Дані поділяються на дві великі групи (рисунок 1.4).

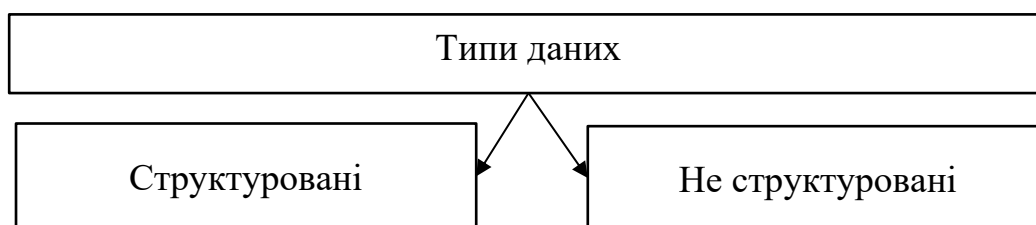


Рисунок 1.4 – Класифікація даних по відповідності структурі

Неструктуровані дані представляють собою набори інформації, які не мають чіткої передбаченої структури перед обробкою. Зазвичай це текстові дані, такі як відкриті відповіді на опитування та бесіди в соціальних мережах, але також можуть включати нетекстові формати, такі як зображення, відео та аудіо. Зараз неструктурована інформація стрімко зростає завдяки розширенню цифрових програм і послуг. Хоча структуровані дані також є важливим ресурсом, неструктуровані дані мають ще більший потенціал для бізнесу, якщо їх правильно проаналізувати. Це може надати значну кількість інформації, яку не можна отримати лише зі статистики та цифр. Зазначено, що від 80% до 90% даних компаній є неструктурованими, і ця кількість постійно зростає. Важливо враховувати, що нарахування великої кількості неструктурованих даних може бути викликом, але з правильними інструментами аналізу та технологіями великих даних це може призвести до важливих відкриттів.

Термін "великі дані" охоплює структуровані, неструктуровані та напівструктуровані дані. Навіть якщо всі три типи даних можуть бути цінними для отримання інсайтів, важливо визначити, який тип даних збирати і коли, а також які саме аспекти аналізувати для досягнення конкретних бізнес-цілей.

Неструктуровані дані, навіть якщо вони містять числа, статистичні дані і факти, часто включають значну кількість тексту або форматування, яке ускладнює їх аналіз. Наприклад, дописи в соціальних мережах можуть містити в собі думки, обговорення тем і рекомендації, але цю інформацію часто складно обробляти масово. Спочатку необхідно витягти та класифікувати конкретні фрагменти інформації, а потім провести аналіз для отримання корисних висновків.

У порівнянні з неструктурованими даними, структуровані дані зазвичай мають числовий характер і їх легко аналізувати. Ці дані організовані в попередньо визначеному структурованому форматі, такому як таблиці Excel чи Google Sheets, де дані вводяться у стандартизовані стовпці та рядки згідно з певними параметрами. Модель структурованих даних розроблена для зручного введення, пошуку, порівняння та вилучення інформації.

Також існують напівструктуровані дані, які, хоча і мають текстовий характер, вільно організовані за категоріями або "мета-тегами". Цю інформацію можна легко розділити на групи, але самі дані в цих групах залишаються неструктурованими.

Прикладом напівструктурованих даних є електронна пошта, де ви можете шукати за папками, такими як "Вхідні", "Надіслані" та "Чернетки", але текст електронного листа в кожній категорії може бути організованим по-різному.

Приклади неструктурованих даних включають юридичні документи, аудіо, чати, відео, зображення, текст на веб-сторінці та багато іншого. Нижче наведено кілька найпоширеніших прикладів неструктурованих даних з якими приходиться зустрічатись кожного дня (рисунок 1.5).

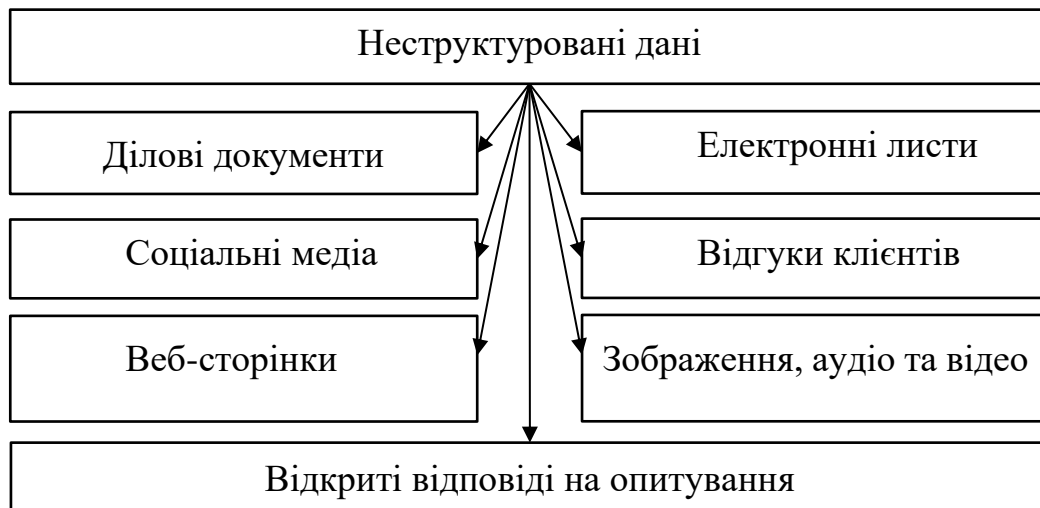


Рисунок 1.5 – Приклади неструктурованих даних

Ділові документи, такі як письмові звіти, юридичні документи та презентації, часто існують у різних форматах: вони можуть бути надруковані на папері, представлені у форматі PDF або, навіть, написані вручну. Деякі з цих документів можуть містити електронні таблиці, зображення або файли XML. Хоча текстові файли можуть мати загальний формат, дані в них не завжди структуровані так, щоб їх можна було легко проаналізувати без використання передових технологій обробки природної мови (NLP). Ці документи містять велику кількість неструктурованих даних, які залишаються невикористаними через те, що їх аналіз вимагає значних зусиль та часу. За допомогою методів аналізу тексту компанії тепер можуть витягати цінну інформацію з цих документів про клієнтів та співробітників і використовувати її для конкурентних досліджень.

Електронні листи, які щодня ми відправляємо великою кількістю, містять значну кількість неструктурованих даних. Навіть якщо електронні листи частково структуровані за категоріями, дані в кожному листі є неструктурованими. Програмне забезпечення для аналізу тексту може швидко сканувати тисячі електронних листів, отримувати інформацію про клієнтів, класифікувати їх за категоріями та направляти до відповідного відділу, а також відстежувати якість обслуговування клієнтів та інші важливі аспекти. За

допомогою аналізу тексту можна навіть визначити, яка мова найкраще підходить для спілкування з клієнтами, легко проаналізувати текст, щоб виявити головні проблеми клієнта всього за кілька хвилин. Наприклад, можна виявити певні теми, які найчастіше згадуються в негативному контексті.

Дані соціальних мереж подібні до електронних листів, але вони масово зростають і представляють собою величезний обсяг неструктурованих даних, що включають ідеї, думки та статистику в реальному часі. Хештеги та інші елементи можуть упорядковувати дані, але повідомлення, що містять ці хештеги, залишаються неструктурованими. Аналіз тексту в соціальних мережах дозволяє вам витягти цінну інформацію з цього обсягу даних, слідкувати за тенденціями та отримувати реальний звіт про думки користувачів.

Відгуки клієнтів, надходячи у різних формах, також містять значний обсяг неструктурованих даних. За допомогою аналізу тексту можна систематизувати інформацію про враження та думки клієнтів, отримуючи повністю збалансоване уявлення про їхню думку. Можна визначити основні проблеми, вносити зміни та відстежувати результати, використовуючи інструменти, такі як аналіз настроїв і хмари слів.

Великий обсяг неструктурованих даних генерується від веб-сторінок, які можуть містити різноманітний контент. Аналіз тексту дозволяє видобути, вилучити та впорядкувати ці дані для пошуку інформації про клієнтів, конкурентів та загальні настрої громадськості. Також можна використовувати машинне навчання для постійного відстеження та порівняння змін на веб-сторінках протягом часу. Всі ці дані можуть бути ефективно оброблені та проаналізовані за допомогою інструментів аналізу тексту, що використовують машинне навчання, щоб автоматично визначати тенденції, виділяти ключові питання та надавати цінні інсайти для бізнесу.

Відкриті опитування, як правило, надають більше глибини та контексту у відповідях респондентів, але їх обробка вимагає уважності. Після збору неструктурованих відповідей опитування, їх можна впорядкувати та проаналізувати за допомогою інструментів бізнес-аналітики, які

використовують машинне навчання для класифікації, аналізу та візуалізації даних.

Зображення, відео та аудіо містять значний обсяг неструктурованих даних, але завдяки новим технологіям, таким як конвертація мовлення в текст та програмне забезпечення розпізнавання облич та об'єктів, стає можливим їх аналіз. Наприклад, для аналізу аудіофайлів можна використовувати перетворення мовлення в текст для подальшого аналізу природною мовою.

Неструктуровані дані, такі як електронні листи, ділові документи та дані соціальних мереж, представляють собою значущий ресурс, якщо їх правильно аналізувати. Технології машинного навчання, такі як обробка природної мови (NLP) та штучний інтелект (AI), дозволяють швидко та точно аналізувати ці дані, виділяючи ключову інформацію та сприяючи у прийнятті обґрунтованих бізнес-рішень.

Важливість неструктурованих даних стає все більш очевидною, і технології машинного навчання відкривають нові можливості для їх ефективного використання в бізнесі. Аналіз таких даних дозволяє автоматизовано та точно отримувати інсайти з великого обсягу неструктурованої інформації.

Інструменти аналізу та обробки неструктурованих даних, такі як MonkeyLearn Studio, вирізняються своєю швидкістю та точністю. Вони використовують техніку машинне навчання, а також обробку природної мови для ефективного автоматизованого аналізу великих обсягів неструктурованих даних. Ці інструменти дозволяють вам отримувати актуальну інформацію в реальному часі, автоматизувати процес впорядкування та класифікації даних, а також підвищити ефективність взаємодії з клієнтами. Крім того, вони стежать за конкурентами, дозволяючи вам виявляти тенденції та впроваджувати інновації на ринку. Застосування інструментів аналізу неструктурованих даних важливо для отримання цінної інформації та прийняття обґрунтованих рішень в умовах зростаючого обсягу даних і швидкозмінюваного бізнес-середовища.

Важливо активно вивчати думки та враження своїх клієнтів, і програмне забезпечення для машинного навчання в цьому може великою мірою допомогти. Зокрема, воно здатне автоматично аналізувати відкриті опитування, електронні листи, а також виявляти відгуки у соціальних мережах, блогах та інших онлайн-джерелах. Такий інструмент може визначати назву компанії, розпізнавати ключові слова відповідно до галузі та проводити аналіз настроїв для визначення думки автора. Шляхом аналізу тисяч публікацій у соціальних мережах або в режимі реального часу, ви можете виявити згадки про ваш бренд та слідкувати за реакціями користувачів. Це дозволяє збирати важливі дані щодо поглядів і відгуків на ваш продукт чи послугу.

Аналіз настроїв на основі аспектів є ще одним корисним методом. Ця техніка розбиває текст на аспекти, такі як функції чи компоненти продукту, і призначає кожному аспекту відповідний настрій (позитивний, негативний або нейтральний). Такий підхід дозволяє здійснити більш глибокий аналіз та розуміння відгуків користувачів. Це не лише дозволяє збирати інформацію про сприйняття вашого бренду, але і допомагає виявляти потенційні проблеми та визначати напрямки для подальших вдосконалень у продукції та маркетингових стратегіях.

Раніше аналіз неструктурованих даних обмежувався системами керування документами, де зберігалися метадані та історія версій, але в основному аналіз проводився вручну. Зараз існує нове покоління інструментів аналізу даних, які використовують передові алгоритми машинного навчання та обробки природної мови для ефективного розбиття неструктурованих даних, забезпечуючи збереження та використання результатів.

Перш ніж приступити до аналізу, важливо визначити кінцеву мету та конкретні цілі. Це допоможе визначити, яку інформацію вам потрібно здобути та який тип неструктурованих даних слід збирати.

Під час збору відповідних даних важливо визначити джерела, з яких вони будуть отримані, чи це соціальні мережі, онлайн-огляди чи опитування. Чистка

даних перед аналізом включає в себе усунення шуму, видалення непотрібної інформації та структурування даних для полегшення їх подальшого аналізу.

Важливим етапом є впровадження технології, такої як інструменти аналізу неструктурованих даних, але також архітектури зберігання даних, таких як бази даних NoSQL, і інструментів візуалізації даних, як Tableau і Google Data Studio. Це допомагає забезпечити ефективний потік даних та надає можливість взагалі використовувати отриману інформацію.

Збираючи всі ці елементи, ви створюєте здатність розуміти та аналізувати неструктуровані дані, забезпечуючи можливість отримання цінної інформації для вашої команди та бізнесу в цілому.

1.2 Технологія машинного навчання

Машинне навчання (ML) – це галузь штучного інтелекту (AI), яка дозволяє комп'ютерам «самонавчатися» на основі навчальних даних і вдосконалюватися з часом без явного програмування. Алгоритми машинного навчання виявляють шаблони в даних та вчаться на їх основі для прогнозування результатів. У порівнянні з традиційним програмуванням, де інженери пишуть вказівки для обробки вхідних даних та визначення бажаного результату, машинне навчання використовує автоматизований процес, де машини самостійно розв'язують завдання на основі досвіду.

У традиційному програмуванні використовуються інструкції типу if-then, визначаючи дії на основі конкретних умов. За відміну від цього, машинне навчання дозволяє машинам вирішувати проблеми на основі минулих спостережень без прямого втручання людини. Терміни "штучний інтелект" і "машинне навчання" часто використовуються як синоніми, проте вони представляють різні концепції. Штучний інтелект включає в себе ширший спектр можливостей, включаючи рішення приймання, навчання новим навичкам

та розв'язання проблем, що вимагають інтелектуального розуміння. Машинне навчання є частиною штучного інтелекту і дозволяє системам навчатися новому на основі даних. Замість того, щоб прямо програмувати алгоритми машинного навчання, можна подати їм приклади позначених даних (навчальні дані). Це допомагає їм визначати шаблони та робити обчислення, основані на навчальних прикладах. Машинне навчання може обробляти великі обсяги даних та досягати вищої точності у порівнянні з традиційними методами. Його використання дозволяє заощадити час і ресурси при вирішенні завдань, таких як вирішення проблем клієнтів, автоматизація процесів та аналіз великих обсягів даних.

Для розуміння принципів роботи машинного навчання необхідно вивчити різноманітні методи та алгоритми, які, в основному, представляють собою набори правил, що використовуються машинами для прийняття рішень. Нижче наведено п'ять найпоширеніших та найбільш використовуваних типів машинного навчання (рисунок 1.6).

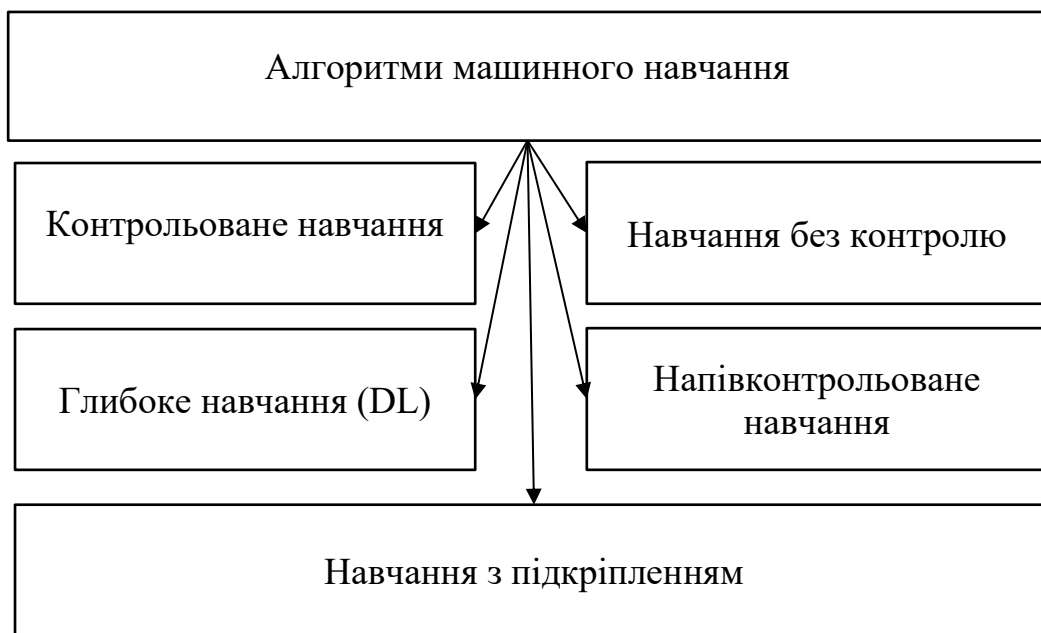


Рисунок 1.6 – Класифікація алгоритмів машинного навчання

Контрольоване навчання. Алгоритми та моделі навчання під наглядом здатні робити прогнози на основі позначених навчальних даних, де кожен

навчальний зразок містить вхідні дані та бажані вихідні дані. У цьому типі навчання алгоритм аналізує ці дані та робить обґрунтовані висновки, щоб приймати рішення для невидимих даних. Це широко використовуваний підхід до машинного навчання, де модель навчається на основі маркованих даних, які вручну позначені тегами, щоб навчити машину розпізнавати та класифікувати певні шаблони. Наприклад, при виявленні спаму модель навчання повинна бути попередньо позначена прикладами електронних листів, які вважаються спамом та тими, що не є спамом. В контрольованому навчанні виділяють два типи завдань: класифікація та регресія. Алгоритми класифікації, такі як опорні векторні машини та наївний Байєс, визначають категорію для вихідних значень. Наприклад, модель аналізу настроїв може автоматично класифікувати дані як позитивні, негативні чи нейтральні. У регресійних завданнях очікується неперервне число, і ця модель використовується для прогнозування числових значень в межах певного діапазону, таких як вартість нерухомості або поширення COVID-19 у певному регіоні.

Навчання без контролю. Алгоритми неконтрольованого навчання виявляють ідеї та зв'язки в немаркованих даних. У цьому випадку моделі отримують вхідні дані, але бажані результати невідомі, тому вони повинні робити висновки на основі непрямих доказів, без конкретних вказівок чи навчання. Моделі не навчаються конкретним "правильним відповідям", тому вони повинні самостійно визначати шаблони. Один із найпоширеніших типів неконтрольованого навчання - це кластеризація, що включає групування схожих даних. Цей метод часто використовується для дослідницького аналізу та може допомагати виявити приховані закономірності чи тенденції. Наприклад, команда маркетингу компанії електронної комерції може використовувати кластеризацію для поліпшення сегментації клієнтів. Маючи набір даних про доходи та витрати, модель машинного навчання може ідентифікувати групи клієнтів із схожою поведінкою. Сегментація дозволяє маркетологам адаптувати стратегії для кожного ключового ринку, надаючи рекламні акції та знижки для клієнтів із

низьким доходом, які активно користуються сайтом, як спосіб винагородження за лояльність та підвищення залученості.

Напівконтрольоване навчання. У напівконтрольованому навчанні навчальні дані розділяються на дві частини: невелику кількість позначених даних і більший набір непозначених даних. Модель використовує позначені дані як вхідні для зроблення висновків щодо непозначених даних, що призводить до більш точних результатів, ніж у звичайних моделях контрольованого навчання. Цей метод набуває популярності, особливо для завдань з великими наборами даних, таких як класифікація зображень. Напівконтрольоване навчання ефективне, не вимагаючи великої кількості позначених даних, що робить його швидким у налаштуванні та економічно вигідним порівняно з методами контрольованого навчання. Це ідеально підходить для підприємств, які опрацьовують величезні обсяги даних.

Навчання з підкріпленням (RL) Навчання з підкріпленням стосується того, як програмний агент (або комп'ютерна програма) повинен діяти в ситуації, щоб максимізувати винагороду. Ці моделі машинного навчання намагаються визначити оптимальний шлях, діючи методом проб і помилок, оскільки навчальні дані відсутні. Агенти вчаться на власних помилках і вибирають дії, що призводять до найкращого рішення або максимальної винагороди. Цей метод широко використовується в робототехніці та іграх, де відсутність навчальних даних не заважає агенту вдосконалювати свої стратегії для досягнення максимального успіху.

Глибоке навчання (DL). Моделі глибокого навчання можуть бути контрольованими, напівконтрольованими або неконтрольованими, або використовувати комбінацію цих підходів. Ці передові алгоритми машинного навчання широко застосовуються технологічними гігантами, такими як Google, Microsoft і Amazon, для впровадження систем та живлення інновацій, таких як самокеровані автомобілі та розумні помічники. Глибоке навчання базується на штучних нейронних мережах (ШНМ) - це тип комп'ютерних систем, які моделюють роботу людського мозку. Алгоритми глибокого навчання, або

нейронні мережі, будуються з численних шарів взаємопов'язаних нейронів, що дозволяє системам працювати разом та поетапно навчатися. Коли модель отримує вхідні дані, такі як зображення, текст, відео чи аудіо, і запитується виконати завдання (наприклад, класифікацію тексту за допомогою машинного навчання), дані проходять через кожен рівень, дозволяючи моделі поступово вивчати завдання. Це аналогічно до еволюції людського мозку, який розвивається з віком та досвідом. Глибоке навчання широко використовується у розпізнаванні зображень, розпізнаванні мови та обробці природної мови (NLP). Моделі глибокого навчання зазвичай ефективніше працюють для вирішення складних завдань та обробки об'ємних даних порівняно з іншими алгоритмами машинного навчання. Однак для їхнього навчання зазвичай потрібні величезні обсяги даних та значний час. Щоб зрозуміти, як працює машинне навчання, важливо розуміти поняття "тег". Наприклад, для навчання моделі розпізнавання зображень ви маркуєте фотографії тварин відповідними тегами. Це відомо як маркування даних. При роботі з аналізом тексту за допомогою машинного навчання ви навчаєте модель, надаючи їй дані для аналізу тексту, і позначаєте їх тегами в залежності від проведеного аналізу.

В найпростішому вигляді процес машинного навчання можна розбити на три етапи:

Етап 1. Подаємо навчальні вхідні дані моделі машинного навчання. На цьому етапі надаємо моделі вхідні дані для навчання. Це можуть бути, наприклад, коментарі клієнтів з соціальних мереж або дані служби підтримки клієнтів.

Етап 2. Позначаємо навчальні дані бажаним результатом. На цьому етапі позначаємо навчальні дані тими результатами, які хочемо отримати в результаті аналізу. У випадку аналізу настроїв, наприклад, вказуємо моделі, чи є кожен коментар або частина даних позитивними, нейтральними чи негативними. Модель перетворює ці дані в текстові вектори, тобто числа, які представляють характеристики даних.

Етап3. Перевіряємо модель на тестових даних. На цьому етапі перевіряємо ефективність моделі, надаючи їй тестові (або невидимі) дані. Алгоритми вивчають встановлювати зв'язки між векторами ознак і тегами, використовуючи зразки, які були позначені тегами вручну. Після цього модель може робити прогнози під час обробки невидимих даних.

Якщо після тестування нова модель відповідає вашим стандартам і критеріям, її можна використовувати для обробки різноманітних нових даних. У випадку, якщо вона не працює достатньо точно, то може знадобитися продовжити процес тренування. Однак важливо пам'ятати, що з часом, зі зміною людської мови та термінології в галузі, модель може вимагати постійного навчання новою інформацією.

DL довела свою ефективність у різних типах підтримки прийняття рішень: підтримка прийняття рішень за допомогою розкриття фінансової інформації [18], прогнозування поведінки процесу [29], розпізнавання текстових емоцій [10] та багато інших. Крім того, у сфері виявлення АОВ DL-моделювання набуло великої популярності протягом останніх двох років кількість академічних досліджень моделей глибокого навчання для виявлення кіберзалякування зростає в геометричній прогресії. Пізніші дослідження показують тенденцію до порівняння більш складних моделей DL із класифікації тексту з досить простими архітектурами. Двонаправлена обробка виявилася успішною в області розпізнавання текстових емоцій [10]. Крім того, ми порівнюємо механізми уваги, спосіб скорочення даних, зберігаючи інформацію про проміжні приховані стани, а не лише про останній прихований стан, з більш простими методами зменшення – глобальними шарами об'єднання. Нарешті, ми досліджуємо, чи може ієрархічне представлення даних і застосування уваги до різних шарів, популярний підхід у класифікації документів, покращити ефективність прогнозування. Як частковий внесок у фінальну ієрархічну частину представляємо ієрархічну мережу уваги псевдоречень – розширення ієрархічної мережі уваги – нещодавній прогрес у класифікації документів.

1.3 Програмні засоби для обміну та аналізу повідомлень

Проблема програмного забезпечення для роботи з соціальними медіа полягає в тому, що вони стикаються з однаковими обмеженнями, оскільки функції, до яких вони мають доступ, різняться в різних соціальних мережах. Це призводить до того, що більшість інструментів для роботи з соціальними мережами не лише надають дуже схожі функції, але і відрізняються в залежності від того, які соціальні мережі вони підтримують. Наприклад, TikTok може надати інший набір аналітичних даних, ніж Facebook, та Instagram може функціонувати відмінно від YouTube. Наприклад, Twitter мав одне з найбільш дозволених API, які використовувалися для розробки додатків управління соціальними мережами. Однак після купівлі його функціональність стала набагато обмеженішою та вимагає великих витрат, що ускладнює доступ до функцій, таких як моніторинг конкурентів та прослуховування соціальних мереж.

Розглянемо та виділимо основні функціональні можливості програмних засобів для роботи та обміну повідомленнями у соціальних мережах.

MonkeyLearn - це ефективна платформа машинного навчання SaaS з набором інструментів для аналізу тексту. Вона дозволяє отримувати статистичні дані та результати в реальному часі на основі різноманітних текстових даних, таких як взаємодія з клієнтами, коментарі в соціальних мережах, онлайн-огляди, електронні листи та інші. За допомогою MonkeyLearn можна легко підключити дані та використовувати попередньо навчені моделі машинного навчання або створити власні моделі без необхідності програмування. Платформа має широкий спектр застосувань, охоплюючи аналіз тексту з різних джерел, таких як соціальні мережі, електронні листи та відгуки. Також важливою перевагою є можливість масштабування для обробки різноманітних обсягів даних, а також різні методи аналізу, такі як аналіз тем, аналіз настроїв і вилучення тексту.

Окрім того, MonkeyLearn Studio надає зручний інтерфейс для розширеного аналізу та візуалізації даних, що дозволяє об'єднувати різні аналізи для спільної роботи. Це робить платформу ефективним інструментом для роботи з текстовими даними та отримання інформації з їх аналізу.

BigML - це розробка, яка має за мету об'єднати всі потоки даних в організації для спрощення співпраці та поліпшення результатів аналізу даних. Вони спеціалізуються у різних галузях, таких як аерокосмічна промисловість, автомобільна промисловість, енергетика, розваги, фінансові послуги, харчова промисловість, охорона здоров'я, Інтернет речей, фармацевтика, транспорт, телекомунікації і багато інших. BigML пропонує інструменти, які вже готові до використання і можуть бути застосовані в різних сферах бізнесу. Користувачі можуть користуватися як попередньо навченими моделями, так і тренувати свої власні, використовуючи методи класифікації, регресії та прогнозування часових рядів.

IBM Watson - це потужна технологія машинного навчання, яка має гнучкість для застосування в більшості галузей і здатність масштабуватися величезними масштабами в хмарному середовищі. Watson Speech-to-Text вважається стандартом галузі для перетворення мовлення в реальному часі на текст, а Watson Language Translator є одним із найкращих інструментів на ринку для перекладу тексту. Watson Studio ідеально підходить для підготовки та аналізу даних і може бути налаштований майже для будь-якої галузі. Крім того, їхній Класифікатор природної мови спрощує створення розширених моделей аналізу тексту в області SaaS.

Google Cloud ML - це рішення SaaS для аналізу зображень і тексту, яке легко інтегрується з усіма інструментами Google, такими як Gmail, Google Sheets, Google Slides, Google Docs і інші. AutoML Natural Language від Google є одним з найвдосконаленіших інструментів аналізу тексту на ринку. AutoML Vision дозволяє автоматизувати процес навчання користувацьких моделей для аналізу зображень, щоб досягти максимальної точності, відповідно до вашого бізнесу.

Бібліотеки машинного навчання з відкритим кодом пропонують колекції готових моделей і компонентів, які розробники можуть використовувати для створення власних додатків замість того, щоб кодувати з нуля. Вони безкоштовні, гнучкі та можуть бути налаштовані відповідно до конкретних потреб. Деякі з найпопулярніших бібліотек з відкритим кодом для машинного навчання наведені на рисунку 1.7.

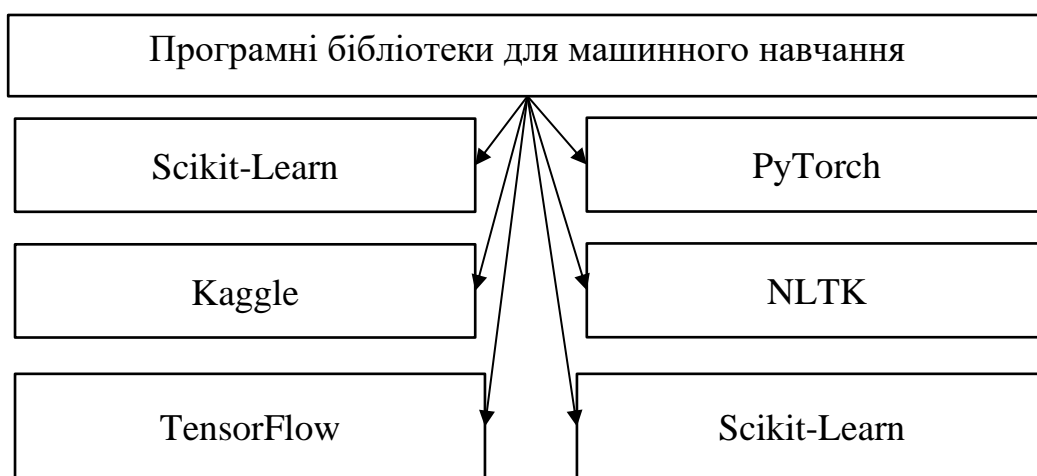


Рисунок 1.7 – Класифікація програмних бібліотек для організації машинного навчання

Scikit-learn – це популярна бібліотека Python і чудовий вибір для початківців у машинному навчанні. Вона вражає своєю простотою використання, надійністю та чудовою документацією. Цю бібліотеку можна використовувати для різних завдань, таких як класифікація, кластеризація та регресія.

PyTorch, розроблений Facebook, є бібліотекою відкритого коду для машинного навчання, заснованою на бібліотеці Torch і орієнтованою на глибоке навчання. Ця бібліотека широко використовується в областях комп'ютерного зору та обробки природної мови, і вона відзначається простотою налаштування, що робить її відмінним вибором для швидкого розв'язання завдань.

Kaggle, запущений понад десять років тому і придбаний Google у 2017 році, визнаний за свою філософію "навчання на практиці". Платформа відома своїми конкурсами, в яких учасники створюють моделі для вирішення реальних проблем. Відвідання цього ресурсу дозволяє швидко навчитися машинному навчанню та створити власну модель.

NLTK (Natural Language Toolkit) - це, можливо, найбільш відома бібліотека для обробки природної мови в Python. За допомогою NLTK можна виконувати завдання, такі як пошук за ключовими словами, токенизація, класифікація, розпізнавання голосу і багато іншого. Бібліотека акцентується на дослідженнях і освіті, і вона надає безліч ресурсів, включаючи набори даних, готові моделі та навчальні посібники, які полегшують початок роботи.

TensorFlow - це бібліотека машинного навчання з відкритим вихідним кодом, розроблена компанією Google для внутрішнього використання та подальшої відкритої ліцензії. З численними ресурсами, посібниками та інструментами TensorFlow допомагає поліпшити навички в машинному навчанні. Ця бібліотека є зручною платформою як для початківців, так і для експертів, і включає все необхідне для створення та навчання моделей машинного навчання, включаючи попередньо побудовані моделі. TensorFlow спеціалізується на глибокому навчанні та підходить для складних проектів з великими обсягами даних. Хоча для його освоєння може знадобитися час та вміння, він має сильну спільноту та ряд посібників для початківців.

1.4 Постановка задач дослідження

В першому розділі кваліфікаційної роботи були досліджені існуючі технології та програмні засоби для опрацювання текстових повідомлень та визначення типів поведінки на основі них. Додактово проаналізовано та виділено

основні переваги використання підходів семантичного аналізу для обробки текстових повідомлень.

Для досягнення поставленої мети необхідно розв'язати наступні задачі.

- провести дослідження технології семантичного аналізу;
- дослідити технології машинного навчання;
- провести аналітичний огляд існуючих програмних засобів обміну та передачі повідомлень;
- проаналізувати існуючі методи та алгоритми визначення агресії в повідомленнях;
- розробити алгоритм виявлення агресивної поведінки у повідомленнях на основі семантичного аналізу;
- реалізувати програмну систему автоматизованого аналізу повідомлень в соціальних мережах з метою виявлення агресивної поведінки.

1.5 Висновки до розділу

Проведено аналітичний огляд методів та алгоритмів семантичного аналізу на основі аналізу їх роботи в сфері обробки текстових повідомлень, що дозволило виділити переваги та недоліки даної технології.

Досліджено сучасні підходи в сфері машинного навчання на основі аналізу алгоритмів навчання штучних нейронних мереж, що дозволило виділити алгоритми для проектування програмної системи аналізу текстових повідомлень в соціальних мережах.

Проаналізовано програмні системи та цифрові бібліотеки для обміну та аналізу текстових повідомлень в соціальних мережах на основі аналізу їх внутрішньої структури та запропонованих типів даних, що дозволило виділити основні модулі програмних додатків даного типу.

2 МЕТОДИ ТА АЛГОРИТМИ ВИЗНАЧЕННЯ АГРЕСІЇ В СОЦІАЛЬНИХ МЕРЕЖАХ

2.1 Алгоритми обробки природних мов

Обробка природної мови (NLP) дозволяє комп'ютерам розуміти людську мову, проаналізувавши граматичну структуру речень та значення окремих слів, а потім використовувати алгоритми для витягнення значень та надання результатів. Це надає сенс людській мові, спрощуючи виконання різних завдань.

Один із найпоширеніших прикладів використання NLP - це віртуальні помічники, такі як Google Assist, Siri та Alexa. NLP розуміє письмові та усні тексти, наприклад, "Привіт, Сірі, де найближча заправка?", і перетворює їх на дані, які легше розуміють машини. Ще одним важливим застосуванням NLP є чат-боти, які допомагають командам підтримки розв'язувати проблеми, розуміючи типові запитання і автоматично відповідаючи. Багато щоденних застосунків, таких як текстові рекомендації під час написання електронних листів, пропозиції перекладу на Facebook чи фільтрація спаму в електронній пошті, використовують технології NLP, хоча користувачі можуть не завжди цього помічати.

Мета обробки природної мови полягає в тому, щоб зробити людську мову, яка є складною, неоднозначною та різноманітною, зрозумілою для машин. Обробка природної мови, штучний інтелект і машинне навчання іноді використовуються як синоніми, але вони є частинами загального терміну штучний інтелект. ШІ включає системи, які імітують когнітивні можливості, охоплюючи від безпілотних автомобілів до систем прогнозування.

Щоб автоматизувати процеси та надавати точні відповіді, використовуючи NLP, необхідне машинне навчання. Цей підхід полягає в застосуванні алгоритмів, які дозволяють машинам навчатися та вдосконалюватися на основі досвіду без явного програмування. Наприклад, чат-боти на основі штучного інтелекту використовують NLP для інтерпретації користувацької мови та

машинного навчання для надання автоматизованих відповідей, враховуючи минулі взаємодії.

Переваги використання технології обробки природніх мов згруповано та наведено на рисунку 2.1.



Рисунок 2.1 – Переваги використання технології обробки природніх мов

Обробка природної мови застосовує дві методики, щоб допомогти комп'ютерам зрозуміти текст (рисунок 2.2).

Семантичний аналіз зосереджений на розумінні значення тексту. По-перше, він досліджує значення кожного окремого слова (лексична семантика), а потім аналізує, як ці слова поєднуються та яке мають значення в контексті. Головні завдання семантичного аналізу включають в себе:

- усунення неоднозначності слів (спроба визначення їх точного значення у конкретному контексті);

- вилучення зв'язків між сутностями (допомагає зрозуміти, як вони пов'язані між собою у тексті).

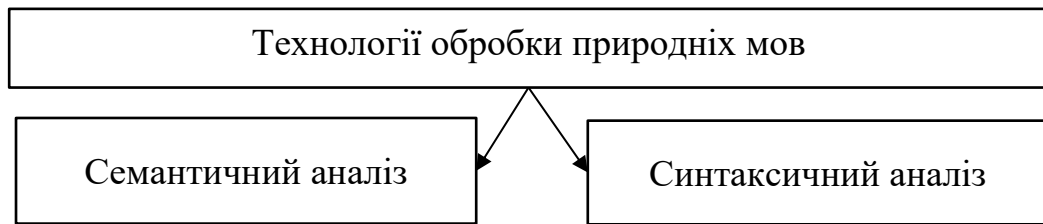


Рисунок 2.2 – Класифікація технологій обробки природніх мов

Синтаксичний аналіз, також відомий як синтаксичний розбір, вивчає текст, застосовуючи основні граматичні правила для визначення структури речення, організації слів та їх взаємозв'язку. Декілька ключових підзадач цього аналізу включають такі аспекти (рисунок 2.3).

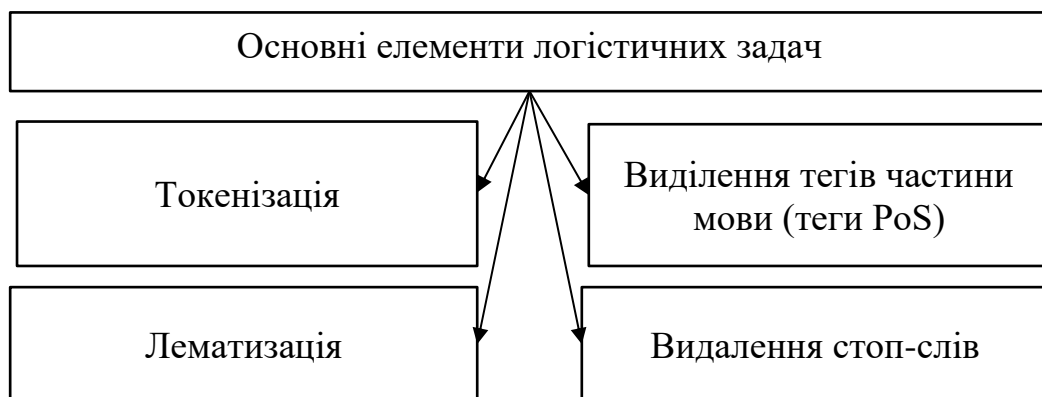


Рисунок 2.4 – Класифікація підзадач синтаксичного аналізу

Токенізація відіграє ключову роль в обробці природної мови, розбиваючи рядок слів на семантично значущі одиниці, які називаються лексемами. Вона може виконувати дві основні функції:

Токенізація речень - розділяє речення в тексті, виділяючи їх як окремі сегменти.

Токенізація слів - розділяє слова в реченні, визначаючи їхні межі. Зазвичай, слова розділяються пробілами, але в деяких випадках може використовуватися високорівнева токенізація для розпізнавання більш складних структур, таких як сполучення слів, наприклад, "Нью-Йорк". Приклад використання токенізації слів для спрощення тексту:

Оригінал: Обслуговування клієнтів не може бути кращим!

Токенізоване: «обслуговування клієнтів» «не» «може» «бути» «кращим».

Такий підхід допомагає розбити текст на окремі лексичні одиниці, що полегшує подальший аналіз та розуміння семантики тексту.

Позначення частини мови (або тегування PoS) включає в себе призначення категорії частини мови кожному слову в тексті. Зазвичай використовуються різні теги PoS, такі як дієслово, прикметник, іменник, займенник, сполучник, прийменник, інтер'єкція тощо. У вищенаведеному прикладі це може виглядати так:

*Обслуговування (дієслово) клієнтів (іменник) не (прислівник) може (дієслово)
бути (дієслово) кращим (прикметник) ! (пунктуація)*

Позначення тегами PoS допомагає встановлювати категорії частин мови для кожного слова у тексті, що полегшує розуміння синтаксичної та семантичної структури речень. Це корисно для аналізу взаємозв'язків між словами і визначення значення тексту.

Грамматика залежностей вивчає спосіб, як слова у реченні пов'язані між собою. Аналізатор залежностей аналізує, як "заголовні слова" взаємодіють та змінюються іншими словами для розуміння синтаксичної структури речення. Приклад конструюється з використанням абстрактних термінальних і нетермінальних вузлів, пов'язаних із словами у тексті.

При розмові або письмі часто використовуємо слова в різних граматичних формах. Щоб полегшити розуміння комп'ютерами, застосовується лематизація та коріння, які перетворюють слова до їхньої основної форми:

Лематизація - відмінювані форми слів зводяться до їхньої лем (основної форми). Наприклад, "ногах" перетворюється у "нога".

Отримання кореня слова – визначення основи слова, яка залишається незмінною при зміні форм. Наприклад, корінь "консульт" у словах "консультувати", "консультант", "консультування", "консультанти".

Слово, як воно з'являється в словнику – його коренева форма – називається лемою. Отже, якщо застосуємо цю лематизацію до речення *«африканські слони мають чотири цвяхи на передніх лапах»*, то результат виглядатиме приблизно так:

*Африканські слони мають чотири цвяхи на передніх ногах =
«африканський», «слон», «мають», «4», «цвях», «на», «передній», «нога»*

Цей приклад корисний, щоб побачити, як лематизація змінює речення, використовуючи його основну форму (наприклад, слово «ногах» було змінено на «нога»). Коли говоримо про корінь, то коренева форма слова називається основою. Корені «обрізають» слова, тому основи слів не завжди можуть бути семантично правильними. Ці процеси полегшують розуміння тексту комп'ютерами, але лематизація зазвичай враховує контекст, вибираючи відповідну лему, тоді як коріння просто оперують окремими словами.

Видалення стоп-слів є важливим етапом у обробці тексту за допомогою методів обробки природної мови. Цей процес включає в себе фільтрацію високочастотних слів, які, взагалі, не несуть семантичної цінності для речення. Серед таких слів можуть бути "котрий", "до", "при", "для", "є", тощо. Наприклад, якщо необхідно класифікувати квитки служби підтримки клієнтів за їх темами, при використанні речення "Привіт, у мене виникла проблема з входом за допомогою мого нового пароля", може бути корисно вилучити стоп-слова, такі

як "привіт", "я", "є", "з", "мій" і залишити тільки слова, що вказують на тему квитка, такі як "проблема", "вхід", "новий", "пароль". Це допомагає зосередитися на суттєвій інформації та поліпшує точність аналізу тексту.

В різних контекстах одне і те саме слово може мати різні значення. Наприклад, слово "книга" може вказувати на об'єкт для читання, але в іншому контексті воно може вказувати на реєстр або набір правил. При розпізнаванні сенсу слова використовують два основних підходи:

- Підхід на основі знань (словниковий підхід). Спроба визначити значення слова, спираючись на словникові визначення та знання про те, як слово використовується в різних ситуаціях. Наприклад, можна використовувати словникові дані, щоб визначити, що "книга" може мати значення роману чи набору правил.

- Контрольований підхід. Базується на алгоритмах обробки природної мови, які навчаються з навчальних даних. Може використовувати контекст, синтаксичні та семантичні ознаки для визначення сенсу слова в конкретному реченні. Навчання моделей на великому обсязі текстових даних дозволяє вирішувати неоднозначність на основі того, як слова вживаються в різних контекстах.

Використання обох підходів може покращити точність розпізнавання сенсу слова в тексті, особливо коли мова йде про обробку природної мови в алгоритмах штучного інтелекту.

Розпізнавання іменованих сутностей – це завдання семантичного аналізу, спрямоване на вилучення та класифікацію різних сутностей з тексту. Сутності можуть включати імена, місця, організації, дати, адреси електронної пошти тощо. Основна мета - ідентифікація конкретних об'єктів або осіб у тексті та присвоєння їм відповідних категорій. Приклад:

Вхідний текст: "Компанія Apple анонсувала новий iPhone, який буде доступний у магазинах з вересня".

В результаті аналізу:

- організація -> Apple;

- продукт -> iPhone;
- дата -> вересень;

Вилучення зв'язків є розширеним завданням, яке йде далі визначення і класифікації іменованих сутностей. Воно фокусується на виявленні семантичних зв'язків між сутностями у тексті. Приклад:

Вхідний текст: "Катерина живе в Тернополі"

Вилучення зв'язків:

- сутність 1 (Катерина);
- сутність 2 (Тернополі);
- відношення (зв'язок): " живе в ".

Це дозволяє зрозуміти, як різні сутності взаємодіють у тексті та визначити семантичні відносини між ними, що є корисним для розуміння контексту та вдосконалення аналізу тексту в глибину.

Класифікація тексту представляє собою процес визначення значення неструктурованого тексту та розподіл його за попередньо визначеними категоріями або тегами. Одним із популярних варіантів цього завдання є аналіз настрою, який спрямований на класифікацію текстів за їх емоційним відтінком. Приклади завдань класифікації тексту наведено в таблиці 2.1.

Таблиця 2.1 – Приклади класифікації тексту

Вхідний текст	Результат аналізу
Цей фільм був дуже захоплюючим та емоційно заворожуючим.	Позитивний настрій
Хочу придбати новий смартфон. Рекомендації?"	Купівля смартфона
Bonjour! Comment ça va?"	Французька мова
Стаття про нові технології в галузі штучного інтелекту.	Штучний інтелект, технології

Завдання класифікації тексту допомагає впорядковувати та аналізувати великі об'єми текстової інформації, спрощуючи розуміння та експлуатацію даних в різних галузях, включаючи мовознавство, аналітику соціальних мереж, та інші.

Обробка природної мови стикається з численними викликами, але однією з основних причин складнощів полягає в неоднозначності людської мови (рисунок 2.5). Інша проблема полягає в тому, що людям важко аналізувати та класифікувати мову коректно.

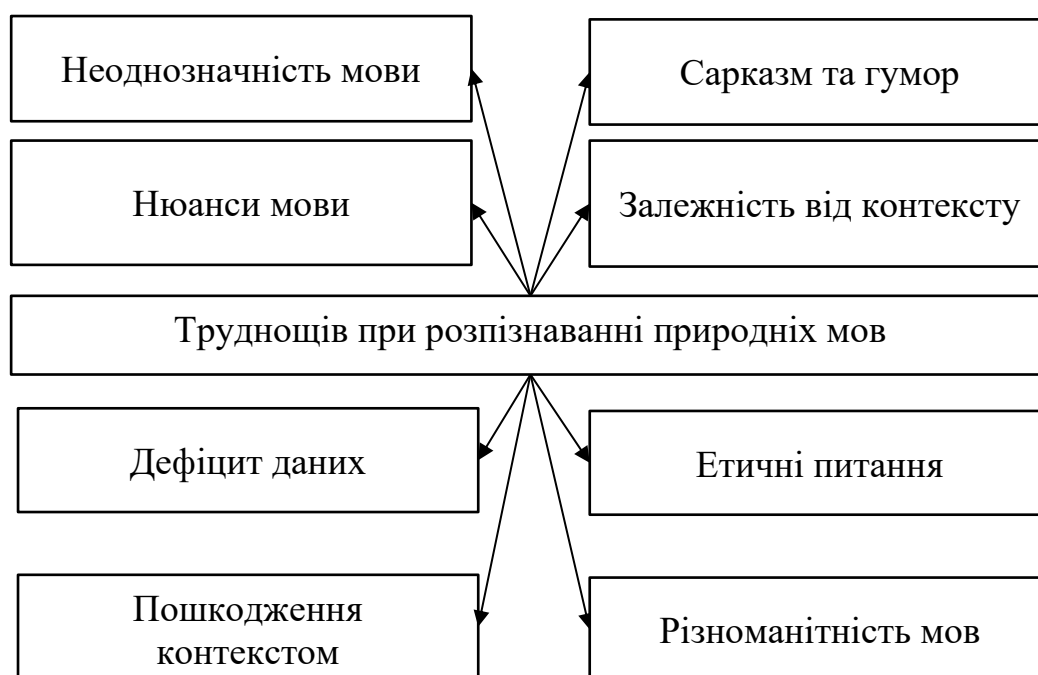


Рисунок 2.5 – Класифікація труднощів при розпізнаванні природних мов

- Неоднозначність мови (слова та фрази можуть мати кілька значень або тлумачень в залежності від контексту).
- Сарказм та гумор (виявлення сарказму чи гумору може бути складним завданням, оскільки вони часто ґрунтуються на відхиленнях від звичайних семантичних правил).
- Нюанси мови (врахування культурних відмінностей, стилістичних особливостей та варіацій у використанні мови).

- Залежність від контексту (розуміння висловлення може змінюватися в залежності від контексту, а не лише від окремих слів).
- Дефіцит даних (В областях, де доступні недостатні дані для навчання моделей, може виникати проблема неадекватності та обмеженої взаємодії).
- Етичні питання (Ризик виникнення викривання, стереотипів та прийняття несправедливих рішень на основі алгоритмів, що працюють з текстом).
- Пошкодження контекстом (зміна сенсу висловлення внаслідок введення нового контексту).
- Різноманітність мов (різні діалекти, мови та сленг можуть створювати труднощі у розумінні тексту).

Наприклад, сарказм може бути важким для машинного розуміння, оскільки вимагає розрізнення протилежного значення висловлення. У розпізнаванні мови важливо враховувати не тільки словникові значення та порядок слів, але й контекст, культурні відмінності та інші нюанси, щоб повноцінно розуміти людську мову. Вирішення цих завдань ускладнюється нестабільністю мови та різноманіттям мовних варіацій. Машинне навчання та високотехнологічні рішення вирішують ці проблеми, проте важливо також враховувати етичні аспекти та уникати виникнення стереотипів та несправедливостей в рішеннях, базованих на тексті.

Обробка природної мови зосереджується на розумінні, аналізі та генерації текстової інформації, роблячи можливим сприйняття та використання великого обсягу даних, які походять із звичайної мови. Алгоритми обробки природної мови відіграють ключову роль у цьому процесі, дозволяючи комп'ютерам взаємодіяти з текстом на подібний до людського способу. Однією з основних завдань є аналіз семантики, або визначення значення тексту. Цей процес розпочинається лексичною семантикою – вивченням значення кожного окремого слова. Алгоритми лексичної аналітики розбирають текст на токени, або окремі одиниці, які є базовими будівельними блоками природної мови.

Токенизація допомагає встановити межі слів, надаючи текстовому потоку структуру та легше використовується для подальшого аналізу.

Усі алгоритми обробки природної мови сприяють розширенню можливостей взаємодії між комп'ютерами та текстовою інформацією, відкриваючи шлях для нових застосувань у різних галузях, від бізнесу та науки до повсякденного життя. Розвиток алгоритмів обробки природної мови слід продовжувати відкривати нові можливості для зростання та розширення впливу цієї технології.

2.2 Алгоритми визначення агресивної поведінки в соціальних мережах

З розвитком соціальних мереж та інтернет-спільнот постає важливе завдання визначення агресії в текстових повідомленнях. Ця проблема стає все актуальнішою в контексті зростання онлайн-комунікацій та потреби забезпечити безпеку інтернет-простору. Алгоритми визначення агресії в тексті відіграють важливу роль у фільтрації та контролі контенту для створення безпечних та позитивних онлайн-середовищ. Однією з головних складнощів у цьому завданні є те, що агресія може виявлятися у різних формах та ступенях виразності. Від простих форм неввічливості до загрозливих висловлень, агресивний контент може мати велику варіативність. Тому алгоритми повинні бути досить гнучкими та адаптованими, щоб впізнавати широкий спектр агресивних висловлень.

Одним із підходів до визначення агресії є використання методів машинного навчання. Моделі можуть бути навчені на великому корпусі текстових даних, де агресія анотована, і виробляти специфічні патерни агресивності. Алгоритми класифікації можуть визначати, чи текст містить агресивний контент, а також визначати його тип та інтенсивність.

Однак, такий підхід має свої недоліки. По-перше, потрібно мати великий та різноманітний набір даних для ефективного навчання моделей. Це може бути

проблемою, оскільки агресія – це часто суб'єктивне поняття, і різні люди можуть сприймати агресію по-різному. Також важливо враховувати контекст, оскільки той самий текст може мати різне значення в залежності від ситуації. Додатково, алгоритми повинні бути здатні враховувати еволюцію мови та нові форми агресивного висловлення, які можуть з'явитися в інтернеті. Це вимагає постійного оновлення та удосконалення моделей. Ще однією проблемою є розпізнавання та аналіз нестандартних виразів, таких як жаргон, сарказм чи іронія. Агресивний контент може бути виражений не тільки прямими загрозами, але і прихованими формами, що робить важким завдання його визначення для алгоритмів. Крім того, етичні питання також важливі в контексті визначення агресії в тексті. Алгоритми повинні бути налаштовані так, щоб уникати цензури або виокремлення неагресивних висловлювань, враховуючи право на свободу слова та відмінності в культурних контекстах.

Онлайн спілкування стало неодмінною частиною сучасного життя, але разом із зростанням цифрової взаємодії зростає і проблема агресивної поведінки в цьому просторі. Для виявлення агресивних ознак у текстових повідомленнях в онлайн спілкуванні використовуються різноманітні алгоритми (рисунок 2.6), які базуються на технологіях обробки природної мови та машинного навчання.

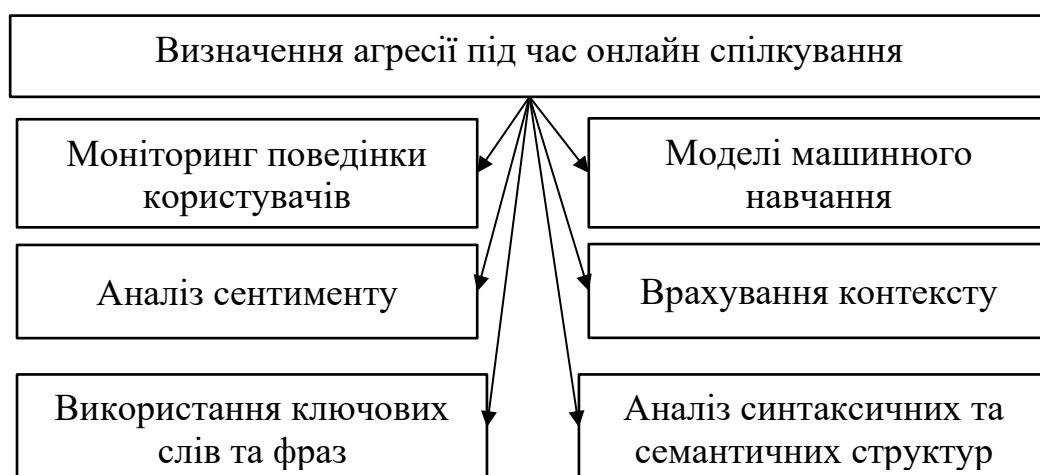


Рисунок 2.6 – Підходи до визначення агресії під час онлайн спілкування

Один із основних підходів - це аналіз настрою, який визначає тон повідомлення – позитивний, нейтральний чи негативний. Агресивні повідомлення часто мають негативний настрій. Алгоритми можуть використовувати методи машинного навчання для класифікації текстів за їхнім емоційним забарвленням. Агресивна поведінка часто супроводжується конкретними словами чи фразами. Алгоритми можуть використовувати словник агресивних слів та фраз для визначення наявності агресії у тексті. Також важливо враховувати контекст, оскільки те саме слово може мати різне значення в різних ситуаціях. Деякі алгоритми використовують аналіз синтаксичних та семантичних структур тексту для виявлення агресії. Вони розглядають взаємозв'язки слів та фраз, щоб зрозуміти, чи вказує текст на агресивну поведінку. Застосування моделей машинного навчання для класифікації текстового контенту – ефективний підхід. Моделі можуть бути навчені на агресивних та неагресивних текстах, розпізнавати патерни агресії та визначати їх в текстових повідомленнях. Важливим є аналіз контексту спілкування. Те, що на перший погляд може здатися агресивним, у певному контексті може мати інше значення. Тому алгоритми повинні бути здатні враховувати обставини та різноманіття ситуацій. Поєднання алгоритмів визначення ознак агресивної поведінки з моніторингом історії поведінки користувачів може поліпшити точність визначення агресії. Постійне вдосконалення системи на основі змін у поведінці може робити її більш адаптованою та ефективною. Необхідно також звертати увагу на етичні аспекти розробки та використання алгоритмів визначення агресивної поведінки, враховуючи право на приватність та свободу висловлення користувачів.

Багато дослідників працювали над текстовими коментарями, зібраними в соціальних мережах з метою визначення образ (нецензурних слів) у коментарях. Для оцінки агресивності повідомлення було визначено п'ять критеріїв:

- відсоток поганих слів у повідомленні;
- поєднання займенника першої особи, поганого слова та займенника другої особи;

- поєднання займенника другої особи з поганим словом;
- поєднання займенника третьої особи з поганим словом;
- поєднання займенника першої особи, поганого слова і займенник третьої особи.

Аналізу піддають функції на основі користувача, функції на основі тексту та функції на основі мережі. Функції, засновані на користувачах, мають на меті описати загальну поведінку користувача (наприклад, хуліган і загальні агресори швидше, ніж звичайні користувачі в публікаціях), функції на основі тексту стосувалися великих літер, вбудовування конкретних слів і позитивних/негативних настроїв. коротким текстом. Мережеві функції були спрямовані на оцінку популярності, взаємності, різниці в силі та впливу користувачів у групі. Було виділено три особливості, які значно покращують виявлення:

- кількість залежностей із власними назвами в однині;
- кількість залежностей, які містять ненормативну лексику;
- кількість залежностей між особистими займенниками та нецензурною лексикою.

Також важливим критерієм оцінки рівня агресії є опис почуттів:

- містять зауваження, критику, сарказм тощо;
- стосуються певних тем (наприклад, політики, злочинів, раси, статі тощо);
- містять лайливі слова.

Автоматичне виявлення кіберзалякування можна використовувати з урахуванням психологічних особливостей користувачів, зокрема характеру, почуттів та емоцій. Особистості користувачів можна визначити за допомогою моделі «Великої п'ятірки» (відкритість до досвіду, сумлінність, екстраверсія, приємність і невротизм) і темної тріади (нарцисизм, макіавеллізм і психопатія), які конкретно стосується злих якостей.

Окрім того виділяють дев'ять ознак для виділення рівня агресивності текстових повідомлень під час онлайн спілкування:

Кількість негативних слів (BW) (як правило реалізується на основі використання спеціалізованих словників негативних вульгарних слів, які використовуються в агресивних цілях, образах і приниженнях).

Кількість «не/ні» (NN) (використання «ні/не» в реченні повністю змінює значення речення з позитивного на негативне або навпаки. Крім того, наявність великої кількості «ні/не» може підкреслити суперечливість коментаря).

Великі літери (U) (логічне значення, яке вказує, чи є коментар великими літерами чи ні. На комп'ютерному жаргоні коментар у верхньому регістрі означає щось кричати. Отже, це можна інтерпретувати як агресію проти когось).

Позитивна/негативна вага коментаря (PW/NW) (функція включає два значення: позитивну та негативну вагу коментаря в діапазоні [0,1]. Для цього виділяють відносний синсет і вагу кожного слова, а потім усереднюють як для позитивних, так і для негативних ваг. Середнє значення вибирається з урахуванням довжини коментаря і, отже, кількості слів).

Використання другої особи (SP) (це логічне значення, яке вказує на наявність або відсутність другої форми однини чи множини в коментарі. Ця функція важлива, оскільки атаки часто супроводжуються з використанням другої особи, таким чином націлюючись на конкретну людину. Подібні функції, як правило, реалізуються за допомогою спеціально створеного словника, що включає дієслова та займенники, що стосуються другої особи).

Наявність погроз (TP), підбурювання до насильства чи самогубства. Логічне значення вказує на наявність погроз, насильства або підбурювання до самогубства в коментарях. Багато негативних коментарів супроводжуються використанням нецензурної лексики чи погроз на кшталт «Я вб'ю тебе» або підбурювання до самогубства «скинутий з мосту», усі вирази вживаються лише в агресивному контексті.

Наявність умов булінгу (KW) (логічне значення вказує на наявність так званих ключових слів кіберзалякування або образ, які використовуються для травмування або нападу на людину (наприклад, ідіот, дурний, ...), але також цільових слів, які самі по собі не мають негативного значення. значення, але в

деяких контекстах, таких як кіберзалякування, їх можна використовувати однаково для образи (наприклад, клоун, свиня, сміття, ...).

Довжина коментаря (L) (функція відображає довжину коментаря в словах. Було помічено, що більшість негативних коментарів складаються з кількох слів, зазвичай не більше трьох.

Вибір цих дев'яти функцій був продиктований як сучасним станом, так і ретельним аналізом реальних коментарів користувачів під час онлайн спілкування. Було виявлено, що мова в Інтернеті груба, повна виразів і слів, що належать до вульгарного жаргону, що залишає мало місця для непорозумінь. В українській мові є багато ідентичних термінів, які використовуються, коли когось словесно атакують. Це призвело до створення списку цих слів, створивши справжній словник нецензурних слів. Вагомість нецензурних слів не обов'язково визначає негативність речення, тому було розроблено іншу функцію, яка враховує вагу, яку слово може мати в реченні, як негативне, так і позитивне. Знову ж таки, було зазначено, що деякі негативні коментарі були написані великими літерами, ніби для імітації більш високого тону голосу. Це спонукало до думки про спосіб відстеження цієї особливості. Ще одна особливість, яка була виділена, це наявність заперечення в агресивності У коментарях в багатьох випадках було зазначено, що сеанси агресії починалися зі слова «ні/не», щоб заперечити жертві. Знову ж таки, присутність другої особи, наприклад: «Я вб'ю тебе» або «Ти повинен померти». Для уточнення словникового запасу «нецензурними словами» було визначено ще два словники з виразами, дуже близькими до агресивної мови неповнолітніх. Перший визначається як вияви підбурювання до насильства з метою побажання чиеїсь смерті. Другі визначаються як вирази, пов'язані з неповнолітніми та образливими висловлюваннями, тісно пов'язаними з тицьками та асонансами з тваринами в образливій формі. Агресія як у залякуванні, так і в більш загальному контексті охоплює ці теми, які поступово розглядалися та застосовувалися для вилучення кожного окремого коментаря.

Загалом, алгоритми визначення ознак агресивної поведінки в онлайн спілкуванні використовують різні підходи, і їх ефективність залежить від комплексного поєднання цих методів та постійного оновлення для врахування змін у мовленні та поведінці користувачів. Розробка алгоритмів визначення агресії в текстових повідомленнях – це складне та багатогранне завдання. Вона потребує поєднання методів машинного навчання, розуміння граматичної та семантичної структури мови, а також постійного оновлення для врахування змін від зростання онлайн-спільнот та змін у мові користувачів. При цьому важливо дотримуватися етичних принципів та забезпечити баланс між свободою висловлення та захистом від агресії в онлайн-середовищі.

2.3 Алгоритм визначення рівня агресії в текстових повідомленнях

В сучасному цифровому світі, орієнтованому на технології та з'єднаному через Інтернет, питання кібербезпеки стають все більш актуальними і важливими. Однією з гострих проблем є кіберагресія, що включає в себе різноманітні форми агресивної діяльності в онлайн просторі. Виявлення кіберагресії стає необхідною задачею для забезпечення безпеки користувачів та попередження можливих наслідків. У цьому есе розглянемо актуальність цієї задачі та способи виявлення кіберагресії. Соціальні мережі, електронна пошта, фінансові транзакції - все це перейшло в онлайн простір, роблячи людей більш вразливими перед кіберзагрозами. Кіберагресія може виявитися у вигляді кібербулінгу, фішингу, атак на системи, чи навіть впливу на громадську думку через фейкові новини. З цього погляду, виявлення кіберагресії стає необхідністю для забезпечення цілісності та безпеки інтернет-середовища.

Соціальні мережі стали основним каналом комунікації для мільярдів користувачів. Однак разом зі зростанням популярності соціальних мереж зросла і кількість випадків кіберагресії. Кібербулінг, образи, фейкові профілі - це лише

кілька проявів агресивної поведінки в цьому контексті. Виявлення кіберагресії в соціальних мережах допомагає вчасно реагувати на негативний вплив та захищати користувачів.

Запропонований алгоритму аналізу та оцінки рівня агресії в текстових повідомленнях буде базуватись на декілької операціях. Отримані тексти розбиваються на токени (слова або фрази), щоб зробити їх більш доступними для подальшого аналізу. Застосування технік токенізації допомагає створити структуровану форму тексту для подальшої обробки. Для кожного токена визначається його частина мови (іменник, дієслово, прикметник тощо). Це важливий крок, оскільки агресія може виявлятися через специфічний вибір слів та їхню граматичну структуру. На основі аналізу частин мови та інших лінгвістичних ознак вилучаються характеристики, які можуть свідчити про агресивний контекст. Це може включати в себе використання образливих слів, інтенсивне використання великих літер або символів. Аналізуються семантичні зв'язки між словами в тексті. Це допомагає виявити контекстуальні зв'язки, які можуть вказувати на загрозовий чи агресивний характер повідомлення. Використовуються побудовані ознаки для класифікації текстів на рівні агресії. Після визначення рівня агресії проводиться аналіз відхилень, який визначає, наскільки результат відхиляється від норми. Це допомагає уникнути неправильних класифікацій і визначити важливість кожної характеристики.

Даний алгоритм використовує декілька типів словників для корекції отриманих результатів:

- словник нецензурних слів;
- словник усталених словосполучень;
- словник вланих назв.

Використання такої кількості словників пов'язано з зручністю створення та заповнення словників даними. Запропонований алгоритм розроблений з використанням механік семантичного аналізу та словників ключових слів з ваговими коефіцієнтами. Послідовність кроків запропонованого алгоритму наведено у вигляді блок-схеми на рисунку 2.7.

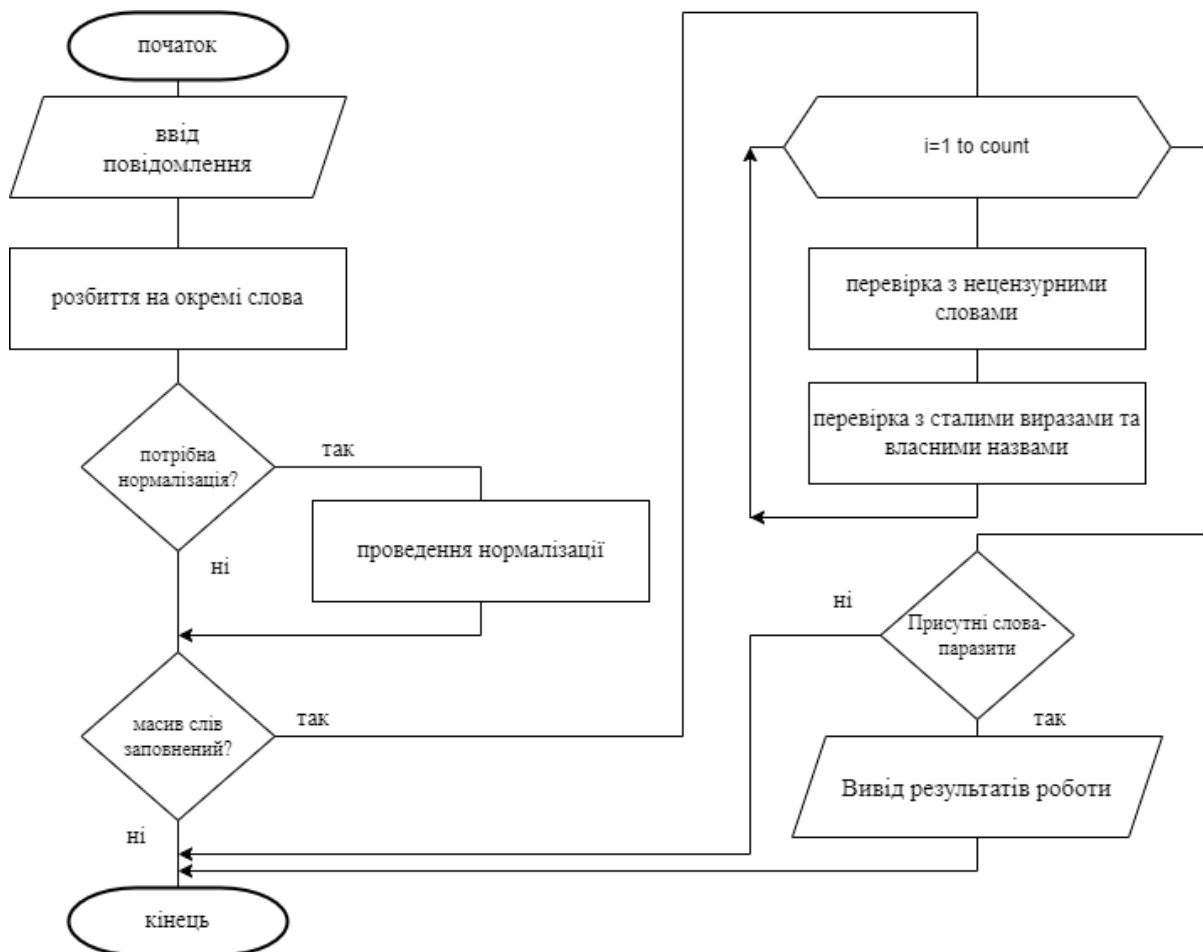


Рисунок 2.7 – Блок-схема алгоритму виявлення рівня агресії в текстових повідомленнях

Запропонований алгоритм складається з чіткої послідовності простих кроків метою яких є отримання оцінки змісту текстового повідомлення на основі аналізу його складових. Використання алгоритмів та підходів до аналізу природніх мов дозволяє провести підготовчі операції та виділити складові елементи вхідного повідомлення. Після чого проводиться послідовна перевірка на наявність серед виділених елементів співпадінь зі словниками, що дасть змогу визначити чи є нецензурні слова або словосполучення-погрози. У випадку якщо відбулось знаходження одного з підозрілих слів, проводиться перевірка на наявність даного слова/словосполучення у словнику власних назв, щоб відкинути випадок некоректного тлумачення аналізованого слова. Запропонований алгоритм у вигляді послідовності команд наведено нижче:

- 1) Ввід текстового повідомлення.

2) Розбиття вхідного повідомлення на окремі слова та визначення їх загальну кількість.

3) Видалення стоп-слів та малоінформативних слів.

4) Нормалізація усіх слів шляхом заміни усіх букв на нижній регістер.

5) Лематизація масиву слів, для отримання слів в основній формі;

6) Перевірка масиву слів з словником нецензурних слів та слів-погроз, та формування масиву підозрілих слів.

7) Якщо не було виявлено жодного підозрілого слова, то переходимо на крок .

8) Перевірка масиву підозрілих слів з елементами масиву власних назв, якщо відбулось співпадіння, то видаляємо дане слово з масиву претендентів.

9) Перевірка масиву підозрілих слів з елементами масиву словосполучень та сталих виразів, якщо відбулось співпадіння, то видаляємо дане слово з масиву претендентів

10) Якщо в масиві підозрілих слів не залишились жодного слова, то переходимо на крок .

11) Проводимо визначення відсотку підозрілих слів відносно цілого повідомлення та визначаємо рівень агресивності повідомлення.

12) Вивід результатів роботи системи.

Опрацювання текстових повідомлень в режимі реального часу стало актуальним завдяки високій ефективності обробки окремого повідомлення для сучасних технічних систем. Однак при використанні конвеєрної обробки даних час, необхідний для опрацювання, збільшується пропорційно кількості повідомлень у черзі. Ще однією важливою характеристикою цього рішення є можливість автоматичної корекції параметрів алгоритмів обробки природньої мови, що розширює функціональні можливості обробки цифрових повідомлень та відкриває нові перспективи для користувачів у сфері аналізу онлайн-повідомлень.

Серед переваг запропонованого алгоритму можна виділити:

- Гнучкість алгоритму, яка забезпечує його адаптивність через можливість коригування параметрів роботи.
- Обробка текстових повідомлень як у режимі обробки поодиноких повідомлень, так і у пакетному режимі для ефективності та оптимізації.
- Автоматизація етапів досліджень, що дозволяє здійснювати аналіз без значного втручання та оптимізує час виконання.
- Здатність працювати в режимі реального часу при аналізі окремих вхідних повідомлень, забезпечуючи оперативну реакцію на нову інформацію.
- Можливість налаштування параметрів роботи для ефективного аналізу повідомлень різної довжини, що робить алгоритм гнучким та універсальним.

Серед недоліків:

- Висока складність реалізації, обумовлена значною кількістю допоміжних алгоритмів, які використовуються на етапах перед обробкою та після неї.
- Необхідність встановлення максимально універсальних початкових параметрів роботи алгоритму, що може виявитися завданням викликливим, оскільки потреби можуть різнитися у різних випадках використання.

2.4 Висновки до розділу

Проведено аланітичних огляд методів та алгоритмів які використовуються при обробці та аналізі текстових повідомлень на природній мові, що дозволило виділити основні етапи при проведенні дослідження текстових повідомлень в системах обміну повідомленнями.

Розроблено алгоритм виділення та підрахунку рівня агресії в текстових повідомлення на основі використання алгоритмів обробки природніх мов, що дозволило спроектувати та проаналізувати структуру програмного додатку аналізу текстови повідомлень.

3 ПРОГРАМНИЙ ДОДАТОК ОЦІНКИ РІВНЯ АГРЕСІЇ В ТЕКСТОВИХ ПОВІДОМЛЕННЯХ

3.1 Структура програмного модуля аналізу текстових повідомлень

В сучасному цифровому світі обробка природної мови стає все більше ключовою складовою для розуміння та взаємодії з текстовою інформацією. Програмні системи, спроектовані для аналізу та оцінки текстових повідомлень, виявляються важливими інструментами для розвитку сучасних технологій. Однією з основних вимог до таких програмних систем є їхній здатність ефективно розуміти та оцінювати смисловий зміст тексту в реальному часі. Такі системи не тільки полегшують завдання обробки інформації, але й розкривають нові можливості для взаємодії та аналізу гігабайтів текстових даних, що генеруються щодня в Інтернеті та в інших сферах. Основним завданням є розпізнавання та інтерпретація текстової інформації з природної мови. Вона оперує не лише структурою речень, а й вивчає контекст, семантику та відтінки значень слів в залежності від контексту. Основні напрямки роботи таких систем включають розпізнавання іменованих сутностей, класифікацію тексту, аналіз настрою, агресії та багато інших аспектів.

Класифікація тексту є одним з ключових завдань програмних систем обробки природних мов. Цей процес полягає в тому, щоб визначити, до якої категорії належить текст (тегування). Наприклад, класифікація може визначити, чи є повідомлення позитивним, негативним чи нейтральним за його настроєм. Для цього система аналізує вживані слова, синтаксичні структури та інші ознаки тексту. Однією з головних проблем, які вирішуються програмними системами для аналізу текстових повідомлень, є визначення семантичного змісту слів у контексті. Слова можуть мати різні значення в залежності від контексту вживання. Таким чином, важливо вирішити проблему визначення семантичного значення слів, враховуючи їх контекстуальні відтінки. Алгоритми визначення агресії в текстових повідомленнях стають все більш актуальними в умовах

розширення соціальних мереж та онлайн-спілкування. Програмні системи виявляють здатність розпізнавати агресивний тон, неприязнь або негативний настрій у текстах, допомагаючи вчасно реагувати на конфліктні ситуації та підтримувати позитивний тон в комунікації. Особливою увагою користуються системи аналізу текстових повідомлень в області кібербезпеки. Аналізуючи величезні потоки текстової інформації, вони можуть виявляти загрози, ризики та аномальні патерни, що сприяє попередженню інцидентів та захисту від кібератак. Загальною тенденцією є розвиток систем, які працюють в режимі реального часу, забезпечуючи швидку та ефективну обробку текстових повідомлень. Це стає можливим завдяки поєднанню потужних алгоритмів машинного навчання та обчислювальних можливостей сучасних комп'ютерів. Програмні системи аналізу та оцінки текстових повідомлень відкривають нові горизонти в обробці природної мови. Вони полегшують взаємодію з інформацією, допомагають вирішувати проблеми спілкування та надають можливості для розширеного використання текстової інформації у різних сферах, від соціальних мереж до кібербезпеки. Важливим є постійний розвиток та вдосконалення таких систем, щоб вони ефективно впроваджувалися в різноманітних сферах життя та відповідали потребам сучасного суспільства.

Вибір мови програмування для написання систем аналізу та оцінки текстових повідомлень – це стратегічне рішення, яке може суттєво вплинути на продуктивність, ефективність та масштабованість системи, що розробляється. Тому перед початком написання програмного коду було проведено аналізу відомих мов програмування з метою вибору оптимальної. Однією з найпоширеніших мов програмування для систем аналізу тексту є Python. Python славиться своєю простотою та читальністю коду, що робить його відмінним вибором для швидкого прототипування та розробки. Велика кількість бібліотек для обробки природної мови у Python, таких як NLTK та SpaCy, забезпечують широкі можливості в галузі аналізу тексту. Відкритий екосистемою Python, що активно розвивається, також спрощує інтеграцію з іншими технологіями.

Для проектів, орієнтованих високу продуктивність, часто використовують мову програмування Java. Java пропонує широкий набір інструментів для створення масштабованих та надійних систем. Завдяки використанню віртуальної машини Java (JVM) програми на Java можуть бути перенесені між різними платформами, що збільшує їх переносимість. Однак у світі у сфері аналізу текстових даних не обійтися без участі мови програмування R. R спеціалізується на статистичному аналізі даних, включаючи текстову інформацію. Багата колекція пакетів, таких як tm (Text Mining) і quanteda роблять R привабливим вибором для дослідників, які займаються аналізом тексту в контексті соціальних досліджень, маркетингу та інших областей. Важливим фактором при виборі мови програмування для систем аналізу тексту є різноманітні інструменти та бібліотеки. У цьому контексті, окрім мов програмування, слід враховувати і середовища розробки. Інтегровані середовища розробки (IDE), такі як PyCharm для Python або Eclipse для Java забезпечують зручність у розробці та налагодженні програм.

На основі проведеного аналізу та оцінки функціональних можливостей, які можуть запропонувати відповідні мови програмування за основу майбутнього проекту була обрана мова програмування Python. Вибір даної мови ґрунтується на тому, що програмні коди на даній мові програмування мають просту структуру, велику кількість готових рішень та дозволяють зосередити свою увагу не на реалізації стандартних елементів, а реалізовувати власні алгоритми та структури даних на основі вже існуючих. Це значно зменшить час написання програмного додатку, спростить процетур у тестування та оптимізації програмного коду, а також надасть можливість інтеграції сучасних цифрових бібліотек в структуру програмного додатку.

Перед процесом програмної реалізації було спроектовано внутрішню архітектуру майбутньої системи аналізу текстових повідомлень. Узагальнену структуру програмного додатку у вигляді взаємопов'язаних елементарних блоків зображено на рисунку 3.1.

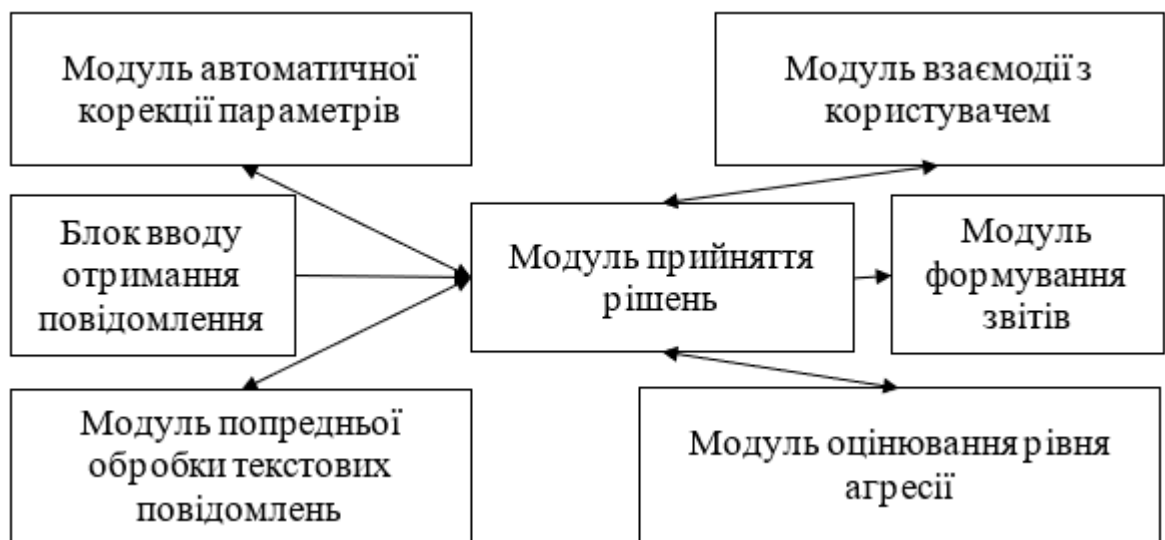


Рисунок 3.1 – Узагальнена структура модуля обробки та аналізу біомедичних зображень

Блок вводу отримання повідомлення. Модуль збору даних є важливою складовою частиною системи обробки текстів, оскільки визначає джерела інформації для подальшого аналізу. В ньому реалізовані функції, які дозволяють використовувати різні методи збору, такі як робота з API соціальних мереж, веб-скрапінг веб-сайтів або підключення до баз даних. Важливою особливістю є адаптивність до різних джерел даних та їхнього формату. Модуль може обробляти різноманітні дані, включаючи текстові повідомлення, коментарі, листи та інші форми текстової інформації.

Модуль попередньої обробки текстових повідомлень. Модуль передпроцесингу виконує ряд завдань для оптимальної підготовки тексту для подальшого аналізу. По-перше, він виконує токенізацію, розбиваючи текст на слова чи речення. Також він проводить видалення стоп-слів, які не несуть семантичного змісту, та виконує лематизацію та визначення частин мови для створення нормалізованого та структурованого тексту. Цей модуль допомагає покращити ефективність аналізу та зменшити кількість шуму у вхідних даних.

Модуль оцінювання рівня агресії. Ключовим компонентом є модуль визначення агресії, який використовує алгоритми аналізу природних мов. Словники ключових слів та правила для оцінки рівня агресії в текстових

повідомленнях. Використовуються функції які реалізують алгоритми аналізу настрою, виявлення ключових слів. Такий підхід дозволяє системі точно визначити тон та емоційний забарвлення тексту, що важливо для виявлення агресії.

Модуль взаємодії з користувачем. Даний модуль створює інтерфейс для легкої взаємодії з системою. Цей модуль надає можливість користувачам вводити параметри аналізу, обирати джерела даних, а також переглядати та аналізувати результати. Інтерактивні функції та можливості налаштування роблять систему більш гнучкою та придатною до використання для широкого спектру користувачів.

Модуль візуалізації результатів. Цей модуль призначений для організації та надання зрозумілого та зручного подання аналізу. Він може використовувати графіки та табличні представлення для представлення результатів аналізу текстових повідомлень. Це допомагає користувачам швидко та легко розуміти ключові аспекти аналізу.

Модуль автоматичної корекції параметрів. Цей модуль додає гнучкість та адаптивність системі. Автоматична корекція параметрів дозволяє системі адаптуватися до змін у вхідних даних та оптимізувати свою роботу. Наприклад, він може реагувати на нові тренди в мовленні або змінювати параметри аналізу в залежності від типу тексту.

Запропонована структура надає можливість налаштування його параметрів в режимі реального часу для більш точного аналізу повідомлень різної довжини. Оброблення текстових повідомлень може відбуватися як у режимі одиночних повідомлень, так і у пакетному режимі, що дозволяє ефективно впоратися з різноманітністю завдань. Автоматизація деяких етапів досліджень робить процес аналізу більш швидким і ефективним для користувача. Важливою особливістю є також адаптивність роботи алгоритму, оскільки можливість корекції його параметрів підвищує гнучкість та допомагає покращити результати обробки. Система включає в себе ряд модулів, які можна розвивати та вдосконалювати з

метою отримання точніших результатів, прискорення швидкості оцінювання та збільшення функціональних можливостей.

Оцінка структури систем на основі інструментів моделювання є ключовим етапом у їх проектуванні та оптимізації. У цьому відношенні використання Unified Modeling Language (UML) – широко визнаної мови моделювання – має велике та визначальне значення. Практика використання UML-діаграм для аналізу структури системи, призначеної для аналізу текстових повідомлень, гарантує візуалізацію, більше розуміння та документування складних взаємозв'язків та компонентів.

Починаючи від класів і компонентів та закінчуючи взаємодіями та розгортанням, ці діаграми стають повноцінним інструментарієм для глибокого аналізу та оптимізації структури системи. Вони виступають невід'ємною складовою частиною процесу проектування та удосконалення сучасних систем автоматизованої мікроскопії.

Щоб отримати повне розуміння взаємодії користувачів із системою визначення рівня агресії в текстових повідомленнях, застосовується діаграма прецедентів. Діаграма прецедентів – це потужний інструмент у галузі моделювання систем, який використовується для візуалізації функціональності системи з погляду її користувачів. Основна мета діаграми прецедентів – надати чітке уявлення про те, як взаємодіють різні елементи системи та які функції вона надає своїм користувачам. Прецедент у даному контексті є конкретною дією або взаємодією, яку користувач (актор) системи виконує з метою досягнення певного результату. Актори являють собою різні ролі або сутності, що взаємодіють із системою. Одним із ключових аспектів використання діаграми прецедентів є визначення всіх можливих варіантів використання системи, що дозволяє розробникам повністю зрозуміти функціональні вимоги та потреби користувачів. Результатом цього аналізу є набір прецедентів, кожен з яких є метою, яку користувач хоче досягти через взаємодію із системою. Діаграми прецедентів часто використовуються на початкових етапах проектування системи для створення загального розуміння її функціональності.

Приклад результатів моделювання на основі використання діаграми прецедентів для програмного модуля обробки та аналізу текстових повідомлень наведено на рисунку 3.2.

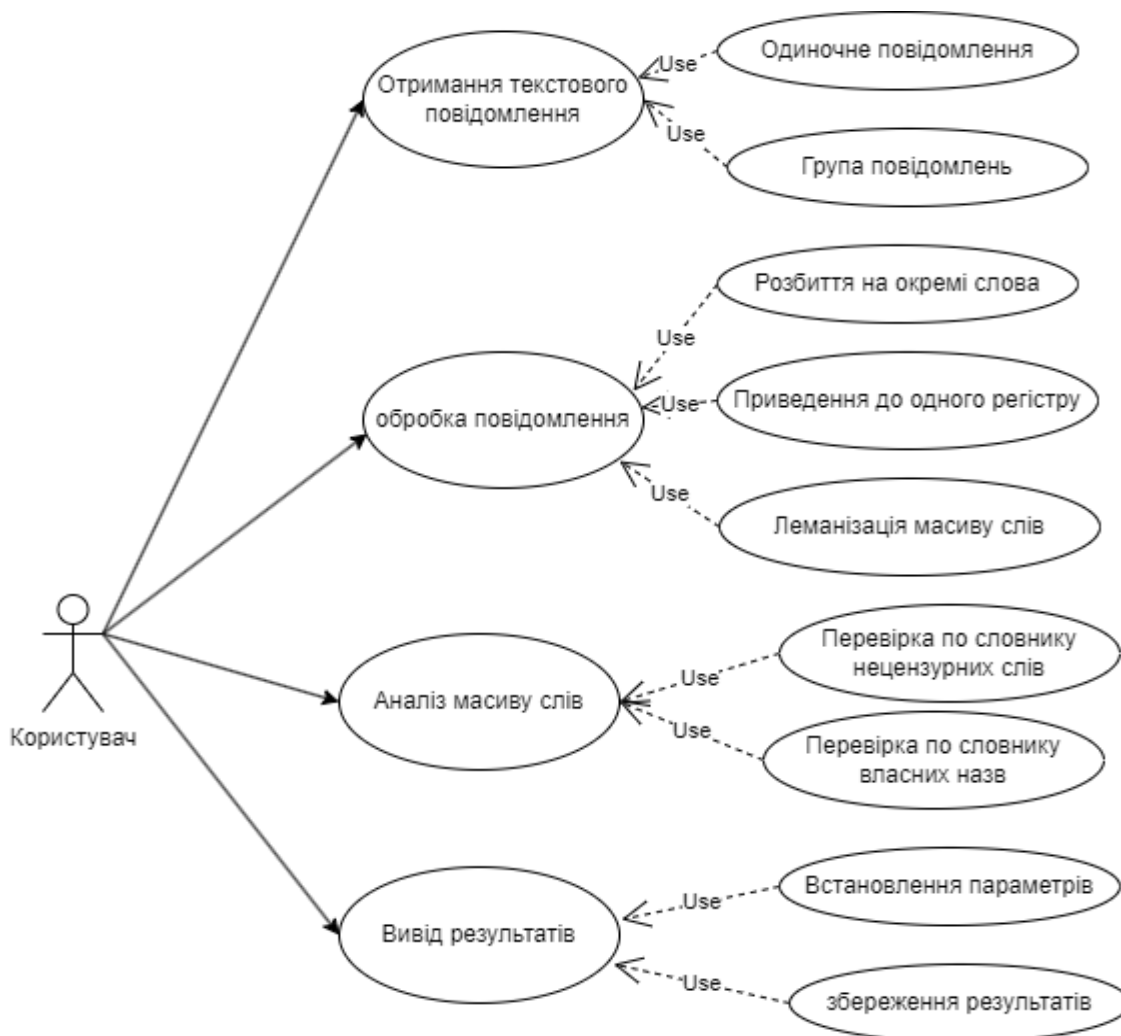


Рисунок 3.2 – Діаграма прецедентів програмного додатку аналізу та оцінки текстових повідомлень

Провівши аналіз біаграми наведеної на рисунку 3.2 можна зробити висновок, що запропонована структура надає для групи акторів «Користувачі» максимальні можливості по взаємодії з усіма функціями програмного додатку. Окремо слід відзначити, що користувачі мають можливість напряду впливати на результати роботи програмної системи шляхом внесення змін в параметри роботи алгоритмів. Проте некоректне внесення значень могло б призвести до

виникнення критичної ситуації та автоматичного завершення роти програми. Щоб уникнути такого розвитку подій в процедуру корекції параметрів були інтегровані максимальні та мінімальні значення, при перетині яких система не дозволяє вносити зміни та повертає значення які були до корекції. Даний механізм дозволяє підвищити стійкість програми до некоректних дій користувачів. Окрім того користувачі можуть формувати загальний вигляд формату виведення результатів роботи програми. Дані налаштування дозволяють вказувати поля які необхідно відображати, а також формат шрифтів якими буде відображатись дана інформація. Це підвищить рівень зручності користування та сприйняття програмного додатку, а також дозволить підняти рівень індивідуальності для кожного з користувачів.

В загальному процес моделювання продемонстрував правильний підхід до розподілу функціональних можливостей для користувачів та повноти доступу до відповідних програмних опцій.

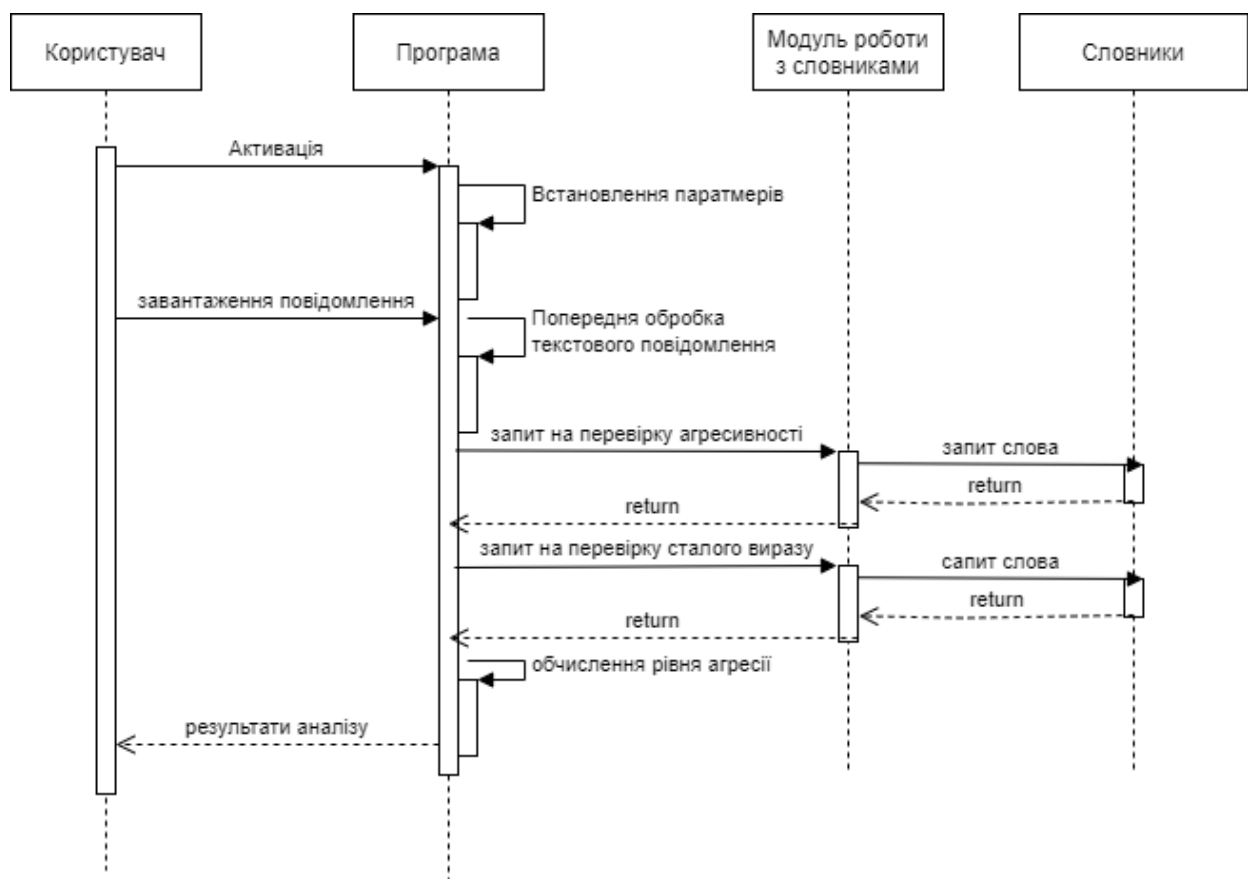


Рисунок 3.3 – Діаграма послідовності програмної системи

Для аналізу взаємодії різних компонентів системи різних сценаріях використання ефективно застосовується діаграма послідовності (рисунок 3.3). Діаграма послідовності є важливим інструментом у галузі моделювання систем. Її основне застосування полягає у візуалізації взаємодії різних об'єктів у системі протягом часу. Вона допомагає краще зрозуміти, як об'єкти обмінюються повідомленнями та які операції вони виконують. Цей інструмент часто використовується для моделювання сценаріїв використання, відображаючи кроки, які виконуються об'єктами у відповідь на різні події. Це спрощує уявлення послідовності дій у межах певного контексту. Аналіз діаграм послідовності може виявити слабкі місця у проектуванні системи, такі як неефективні процеси чи складні ланцюжки взаємодії. Це надає можливість оптимізувати процеси та підвищити ефективність системи. Діаграми послідовності можуть бути основою розробки коду, особливо у об'єктно-орієнтованих системах. Вони надають чітке уявлення про те, які методи викликаються і коли, полегшуючи процес написання коду.

На діаграмі відображено функціонування всіх внутрішніх процесів, що були вибрані та включені в загальну структуру програмного модуля. Проведений аналіз зібраних даних підтвердив, що всі внутрішні процеси взаємодіють без конфліктів, ефективно виконують операції відповідно до обраних сценаріїв, а передача даних відбувається у чіткій послідовності, забезпечуючи повний цикл виконання завдань. Окремі модулі взаємодіють, дотримуючись наперед визначених алгоритмів та обмінюючись інформацією відповідно до вибраних структур. Це створює можливість для здійснення процесу дослідження та інтеграції програмних елементів автоматизації за наперед обраними правилами та параметрами роботи.

Підсумовуючи результати проведеного етапу модулювання можна зробити висновки, що обрана стратегія використання модульного підходу до проектування внутрішньої архітектури програмного додатку обробки та аналізу текстових повідомлень є вірною. Елементи системи взаємодіють за наперед продуманими сценаріями, обмін інформацією відбувається на основі

розроблених структур даних, що дозволяє проводити даний процес контрольовано та без можливих помилок через втрату частини інформації. Користувачам надано доступ до функціоналу який буде відображати основні можливості по виконанні поставлених завдань та дозволить контролювати роботу програмного додатку на усіх етапах роботи. Під час проектування в програму було закладено елементи автоматичної корекції введених результатів, що дозволило організувати захист системи від помилкових дій користувачів під час експлуатації програмного додатку. Усі проведені процедури моделювання підтверджені на основі діаграм які наведені в кваліфікаційній роботі.

Для підвищення рівня зручності використання програмного додатку було розроблено графічний інтерфейс користувача. Дизайн програми було створено на основі результатів порівняльного аналізу програмних систем даного класу. Окрім того дизайн розроблявся з використанням стандартних елементів управління, які використовуються у сучасних офісних програмах. Це було зроблено у тим розрахунком, що звичні елементи управління та дизайну дозволять новим користувачам швидше адаптуватись до роботи з програмною системою, що зменшить час навчання та освоєння. Під час проектування дизайнуна головному вікні були встановлені керуючі елементи для швидкого доступу до найбільш використовуваних функцій програмного додатку, це було зроблено з метою зменшення час доступу до функціоналі програмного додатку. З іншої сторони усі елементи які відповідають за другорядні ункції були приховані, щоб зменшити візуальне навантаження на головне вікно програми, а також збільшити корисну площу головного вікна програмного додатку. Окрім того під час проектування дизайну вікон програмного додатку були враховані сучасні тенденції по прозорості та проектування графічних інтерфейсів користувача для програмних систем загального керування. Розроблений дизайн графічного інтрерфейсу користувача програмного додатку аналізу та оцінки рівня агресії на основі тектових онлайн повідомлень наведено на рисунку 3.5.

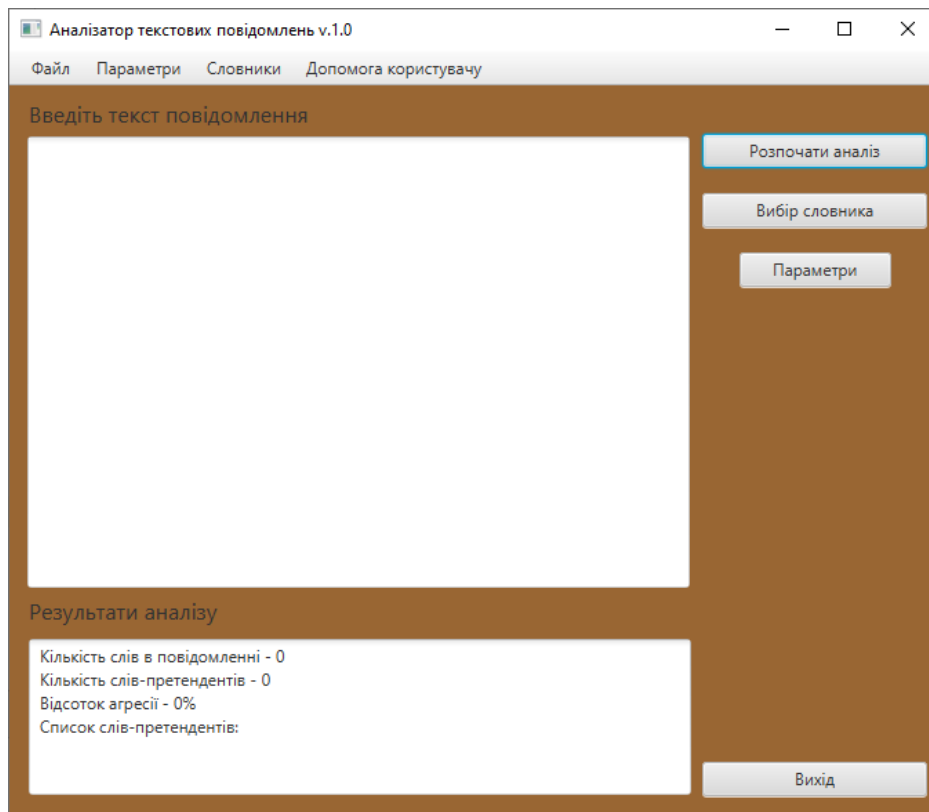


Рисунок 3.5 – Головне вікно програми аналізу та оцінки рівня агресії на основі текстових онлайн повідомлень

На рисунку 3.5 можна побачити відображення головного вікна програмної системи програмного додатку обробки текстових повідомлень. Сам дизайн розділений на декілька окремих областей. В центральній області розташовані два вікна для вводу/виводу текстової інформації. У верхньому вікні буде відображатися текст повідомлення, яке необхідно проаналізувати. Нижнє вікно призначене для виводу інформації про стан процесу аналізу. Зокрема, там буде виводитись поточна інформація про кількість слів у вхідному повідомленні, кількість слів, які були визнані таким, що несуть загрозу, загальний відсоток агресивності цілого повідомлення, а також перелік слів-претендентів які були додані у масив як такі що несуть загрозу. Окрім того для всіх слів-претендентів буде виводитись інформація про їх аналіз, рівень загрози або хибність їх вибору. В верхній частині головного вікна програмної системи розташоване системне меню програмного додатку, яке надає користувачам можливість внести корективи у внутрішні налаштування роботи програмного

додатку, а також проводити маніпуляції з запрограмованими словниками. Користувач може створювати, редагувати та обирати словники для більшої оптимізації процесу аналізу вхідного повідомлення. Третьою візуальною областю на головному вікні програмного додатку є область швидких активаторів функцій програмного додатку. Вони дозволяють користувачам в один клік активувати відповідні функції та розпочати роботу з програмою.

Запропонований дизайн базується на використанні актуальної бази візуальних елементів та сучасних тенденціях розробки графічних дизайнів програмних додатків. Тому його можна використати при реалізації програмного додатку для виконання поставлених перед розробником завдань.

Ще однією умовою спрощення процесу користування програмним додатком є впровадження підходу мінімальної взаємодії користувача з програмою. Це забезпечить швидшу роботу програми, оскільки відпадає потреба очікувати на команди користувача, а також дозволяє створювати сценарії для автоматичної роботи. Приклад послідовності дій користувача при роботі в режимі автоматизації відображено на рисунку 3.7:



Рисунок 3.7 – Послідовність дій користувача при використанні розробленого програмного додатку аналізу текстових повідомлень

Основними функціями користувача, які необхідно виконати для роботи з програмним додатком є встановлення параметрів роботи та при необхідності внесення змін в при візуальному аналізі отриманих результатів. Даний підхід дозволяє мінімізувати вплив користувача на процес обробки повідомлення, що в свою чергу призведе до підвищення рівня об'єктивності оцінювання.

3.2 Функції обробки природніх мов

Проведений аналізу сучасних мов програмування показав перспективу використання мови програмування Python та його бібліотек для реалізації програмного додатку по обробці текстових повідомлень на природній мові. Python славиться своєю лаконічністю та зручністю використання, що робить його ідеальним вибором для обробки текстової інформації. Синтаксис Python є легким для вивчення та зрозуміння, що дозволяє швидко розробляти та тестувати алгоритми обробки тексту. Удосконалений і читабельний код сприяє швидкій реалізації та розширенню функціоналу. Python має багатий екосистем бібліотек, призначених саме для роботи з текстом. Наприклад, бібліотека NLTK (Natural Language Toolkit) забезпечує широкі можливості для роботи з текстом, включаючи токенізацію, стемінг та лематизацію. SpaCy – ще одна потужна бібліотека, яка дозволяє виконувати розпізнавання іменованих сутностей та синтаксичний аналіз. Мова Python чудово інтегрується з іншими інструментами обробки природної мови, такими як Stanford NLP, WordNet та іншими. Це дозволяє розширювати можливості програм за допомогою зовнішніх ресурсів та збільшує точність результатів. Python також відзначається розширеними можливостями візуалізації даних. Бібліотеки, такі як Matplotlib та Seaborn, дозволяють швидко та ефективно відображати результати обробки текстових даних у графіках та діаграмах. Це важливо для зрозуміння та подальшого використання отриманих результатів. Дані переваги дозволяють визначити мову Python як надійний інструмент для обробки текстових повідомлень завдяки своїй простоті, зручності та багатій функціональності.

В напрямку обробки природніх мов бібліотека NLTK (Natural Language Toolkit) визначається як один з ключових інструментів для реалізації завдань. Це насправді не лише набір інструментів, але і повноцінна екосистема, що відкриває великі перспективи для розуміння та обробки людської мови. NLTK надає широкий спектр інструментів для обробки природної мови, починаючи з базових

операцій, таких як токенізація та стемінг, і закінчуючи більш складними алгоритмами, такими як частотний аналіз та морфологічний аналіз. Можливості включають в себе роботу з текстами, класифікацію, переклад, синтаксичний та семантичний аналіз. Токенізація, процес розбиття тексту на слова чи фрази, є основоположним етапом в обробці природної мови. NLTK надає ефективні інструменти для цієї операції, допомагаючи відділити текст на значущі компоненти. Стемінг, в свою чергу, дозволяє зводити слова до їхнього кореня, що спрощує подальший аналіз. NLTK створює зручний інтерфейс для вивчення частотного аналізу та виявлення колокацій, тобто словесних сполучень, які часто зустрічаються разом. Це дозволяє здійснювати глибший аналіз контексту та виявляти взаємозв'язки між словами. NLTK активно використовується в академічних дослідженнях та освітніх програмах, сприяючи розширенню знань у галузі обробки природних мов. Збірні дані та корпуси, які постачає NLTK, відіграють важливу роль у вивченні різних аспектів мови та вдосконаленні методів її аналізу. Більше того, NLTK знаходить широке застосування в промисловості. Від аналізу відгуків у сфері бізнесу до розробки інтелектуальних особистих асистентів – можливості NLTK є різноманітними та вражаючими. Саме ці фактори були основними при виборі бібліотеки NLTK як основної при програмній реалізації програмного додатку обробки та оцінки рівня агресії на основі текстових повідомлень.

Реалізація програмного додатку була розділена на декілька етапів, які дозволяють проводити паралельну розробку та тестування усіх модулів системи. Розглянемо реалізацію окремих елементів програмних структур. Для отримання набору слів які формують повідомлення використовується функція для токенізації `word_tokenize()`. Функція `word_tokenize()` з бібліотеки NLTK призначена для токенізації тексту на окремі слова (токени). Основні принципи її роботи включають наступне:

- Функція розділяє текст на токени на основі пробілів. Кожен розділений фрагмент стає окремим словом (токеном). Пробіли, табуляції та символи нового рядка використовуються як роздільники.

- Функція може розпізнавати різні знаки пунктуації та враховувати їх при токенизації. Наприклад, крапки, коми, лапки тощо.

- `word_tokenize()` може враховувати спеціальні випадки, такі як скорочення, апострофи та інші. Наприклад, "don't" буде розглядатися як одне слово, а не як два окремі токени.

- Функція враховує складні випадки, такі як числа, аббревіатури та інші спеціальні символи. Вона старається розрізнити окремі слова в рядках, де це може бути неочевидним.

- NLTK може використовувати моделі мови для покращення точності токенизації. Вони можуть бути навчені на великих корпусах тексту та враховувати специфіку мови чи жанру.

Загалом, функція `word_tokenize()` намагається забезпечити якнайточнішу та універсальну токенизацію для різноманітних текстів. Приклад використання функції наведено нижче:

```
from nltk.tokenize import word_tokenize
inputText = "Текст для тесту"
outTokens = word_tokenize(inputText)
```

інним завданням яке необхідно вирішити при проведенні аналізу та обробки вхідного текстового повідомлення є аналізу частоти повторення слів, для формування словників. Дане завдання вирішється за допомогою наступного коду:

```
inputText = "Текст для тесту"
outTokens = word_tokenize(inputText)
stop_words = set(stopwords.words("ukraine"))
filtered_tokens = [word for word in tokens if word.lower() not in
stop_words]
freq_dist = FreqDist(filtered_tokens)
sorted_freq_dist = sorted(freq_dist.items(), key=lambda x: x[1],
reverse=True)
```

В даному програмному кодї використовуються ряд додаткових класів, серед яких `FreqDist`. `FreqDist` - це клас з бібліотеки `NLTK`, який використовується для обробки частотних розподілів слів в тексті. Він дозволяє легко визначати частоту кожного слова в текстовому корпусі та виконувати різні операції з цими частотами. Основні функціональні можливості `FreqDist`, він може обчислювати кількість входжень кожного слова в тексті. Визначення найбільш часто вживаних слів: Використовуючи метод `most_common()`, ви можете отримати список найбільш часто вживаних слів та їхніх частот. Ну і при необхідності візуалізації результатів можна побудувати графік, що відображає частотний розподіл слів у тексті.

Окрім того, для зменшення аналізованих даних в програмному кодї видаляються малоінформативні стоп-слова. Це реалізується за допомогою механізму `stopwords.words`. `Stopwords.words` - це список стоп-слів, який включений в бібліотеку `NLTK` і містить слова, які часто зустрічаються в текстах і, як правило, не несуть значущої інформації для аналізу тексту. Використання стоп-слів у текстовому аналізі може допомогти покращити точність та ефективність алгоритмів, так як вони дозволяють ігнорувати загальнопоширені, але маловажливі слова. Важливо відзначити, що список стоп-слів може змінюватися в залежності від мови. В прикладі вище використовується список стоп-слів для англійської мови. Якщо вам потрібно працювати з текстом іншою мовою, вам слід встановити відповідний мовний пакет та скористатися відповідним списком стоп-слів. Для завантаження списку стоп-слів для конкретної мови використовується функція `stopwords.words("мова")`, де "мова" - це код мови (наприклад, "english" для англійської мови, "ukraine" для української).

Приведені приклади проілюстрували ростоту та зручність використання мовипрограмування `Python` та її цифрових бібліотек в процесі створення програмних додатків дл обрбки природніх мов. Більшість рутинних операцій в них вже реалізовано та можуть викликатись за допомогою декількох стрічок коду, що робить процес написання програми швидким та зручним.

3.3 Тестування та аналіз реалізованої програми

Завершальний етап розробки програмного додатку аналізу текстових повідомлень є етап тестування. Для успішного його проведення було обрано робочу станцію з параметрами які відповідають середньому ціновому діапазону та можуть використовуватись як для дому так і в офісних приміщеннях. Завдання компютерів такого типу як правило обмежуються виокремленням офісних програм, роботи в мережі інтернет, а також для використання різних соціальних мереж та меседжерів. Параметри обраної робочої станції наведено на рисунку 3.8

Назва параметра	Значення
Процесор (CPU):	Intel Core i7-11700K
Оперативна пам'ять (RAM):	Corsair Vengeance LPX 32GB DDR4 3200MHz
Жорсткий диск:	Samsung 970 EVO Plus 1TB NVMe SSD
Графічний процесор (GPU):	NVIDIA GeForce RTX 2080
Операційна система:	Windows 10 Pro
Материнська плата:	ASUS ROG Strix Z590-E Gamin
Бездротові технології:	Wi-Fi 6 (802.11ax), Bluetooth 5.2
Порти та роз'єми:	USB 3.2 Gen 2, Thunderbolt 4, HDMI, DisplayPort, Ethernet
Корпус:	NZXT H510 Mid Tower
Блок живлення:	EVGA SuperNOVA 750 G5, 80 Plus Gold

Рисунок 3.8 – Параметри робочої станції для тестування програмного додатку

Для повноти оцінки програмного додатку було обрано набори повідомлень з мережі facebook які часто зустрічаються в українському сегменті даної соціальної мережі. Повідомлення обирались за принципом поширеності та рівня агресії які в них закладав автор. Повідомлення були згруповані по різних групах:

Повідомлення без агресії – група повідомлень які використовуються користувачами при звичайному спілкуванні, передають інформацію та мають дружній характер.

Повідомлення з прихованою агресією, для даної групи повідомлень агресія не задається у прямому вигляді, як правило це можуть бути жартівливі повідомлення або саркастичні.

Повідомлення з псевдо агресією – це повідомлення які містять нецензурні слова або слова агресивного характеру які входять в власні назви або є частиною сталих виразів. Тобто в даній ситуації просто розуміються не за прямим значенням написаного слова.

Агресивні повідомлення. В даній групі зібрані повідомлення які містять текст з прямою агресією, нецензурними словами або спонукають до неправильних дій.

Розглянемо декілька прикладів аналізу текстових повідомлень. Наприклад на рисунку 3.9 наведено результат аналізу звичайного повідомлення, яке містить дружній характер та є звичайним при спілкуванні. Як видно з оцінки обраного повідомлення, то рівень агресії 0% і кількість підозрілих слів рівна 0. Тобто повідомлення не містить жодних слів чи словосполучень, які входять в словники нецензурних слів або сталих виразів.

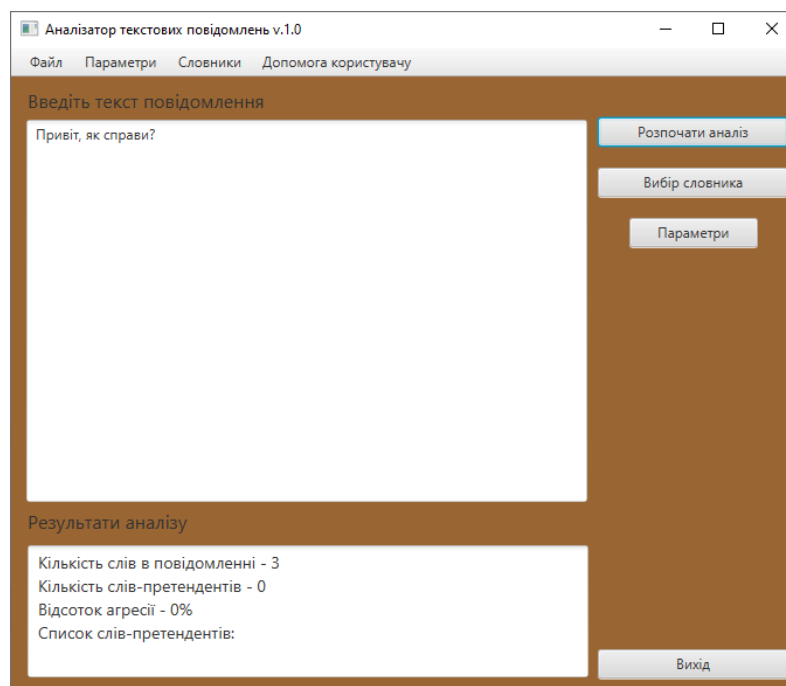


Рисунок 3.9 – Приклад аналізу повідомлення з групи «Повідомлення без агресії»

Наступний приклад проілюструє результати обробки повідомлення з псевдо агресивним змістом (рисунок 3.10). Як видно з результатів оцінки вхідного повідомлення в ньому було виділено одне слово-претендент яке є в словнику нецензурних слів. Проте після додаткового аналізу дане слово виявилось частиною сталого виразу, який хоч і може мати агресивний характер, проте більше використовується для повчання, тому система на основі отриманих даних та аналізу рівня агресії конкретного сталого виразу оцінила загальний рівень агресії як нульовий

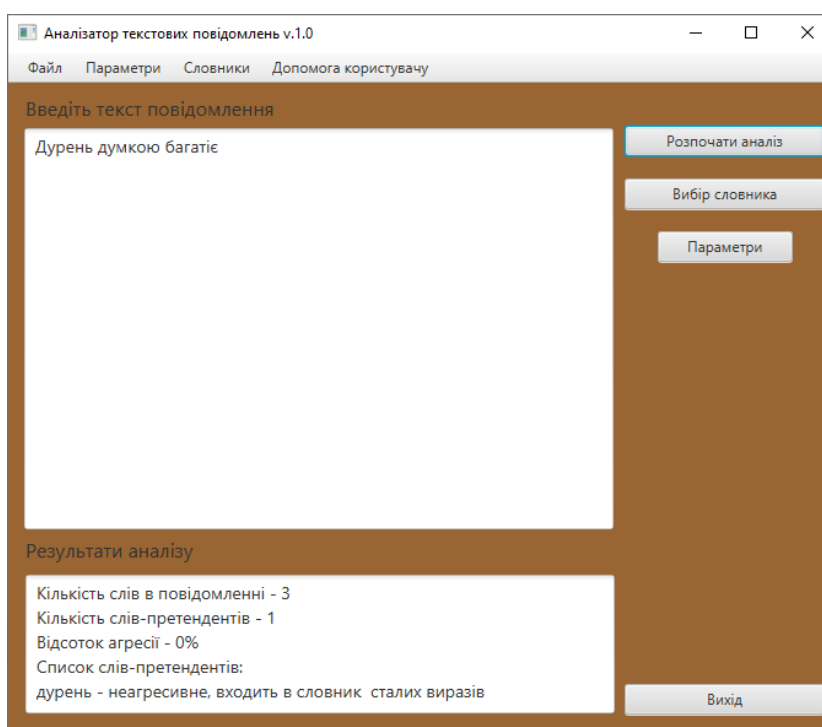


Рисунок 3.10 – Приклад аналізу повідомлення з групи «Повідомлення з псевдо агресією»

В наступному прикладі проілюстровано результати аналізу повідомлення з групи «Агресивні повідомлення» (рисунок 3.11). Під час аналізу було виділено два слова претендента. Перше слово було визначене як напівобразливе, оскільки є прекметником та може використовуватись в цілком дружньому контексті. А друге слово є повністю нецензурним та в даному випадку не входить в склад

сталого виразу. Тому система однозначно прийняла рішення, що дане повідомлення несе агресивний зміст.

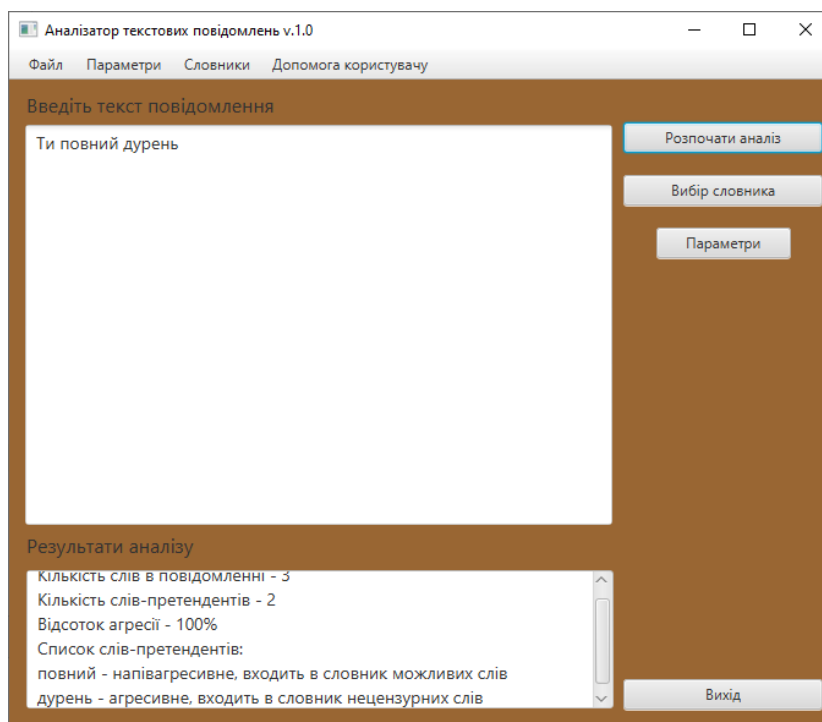


Рисунок 3.11 – Приклад аналізу повідомлення з групи «Агресивні повідомлення»

Окрім того в роботі було проведено аналіз ймовірностей агресивності повідомлення відносно його довжини. Результати дослідження були узагальнені та наведені на рисунку 3.12



Рисунок 3.11 – Ймовірність агресивності повідомлення відносно його довжини

Як показують отримані результати, в основному короткі повідомлення несуть в собі агресивний сенс. Максимальна кількість агресивних повідомлень має довжину 3-4 слова і чим довше повідомлення тим ймовірність його агресивності зменшується. Це можна пояснити тим фактором, що коли людина перебуває в стані агресії їй важче формулювати думки, тому писати довгі логічні послідовності слів не получається.

3.4 Висновки до розділу

Спроектовано та проведено моделювання структури програмного додатку аналізу та оцінки рівня агресії текстових повідомлень на основі алгоритмів обробки природньої мови, що дозволило реалізувати програмний додаток опрацювання текстових повідомлень.

Проведено програмну реалізацію програмного додатку опрацювання текстових повідомлень на основі запропонованого алгоритму оцінки рівня агресії та запропонованої внутрішньої архітектури програмного додатку, що дозволило провести тестування різнотипних текстових повідомлень на визначення рівня агресії.

ВИСНОВКИ

На основі здійсненого аналізу існуючих систем обміну та оцінки повідомлень, алгоритмів обробки природніх мов, а також запропонованому алгоритмові та структурі програмної системи можна зробити такі висновки:

1. Проведено аналітичний огляд методів та алгоритмів семантичного аналізу на основі аналізу їх роботи в сфері обробки текстових повідомлень, що дозволило виділити переваги та недоліки даної технології.

2. Досліджено сучасні підходи в сфері машинного навчання на основі аналізу алгоритмів навчання штучних нейронних мереж, що дозволило виділити алгоритми для проектування програмної системи аналізу текстових повідомлень.

3. Проаналізовано програмні системи та цифрові бібліотеки для обміну та аналізу текстових повідомлень в соціальних мережах на основі аналізу їх внутрішньої структури та запропонованих типів даних, що дозволило виділити основні модулі програмних додатків даного типу.

4. Проведено аналітичний огляд методів та алгоритмів які використовуються при обробці та аналізі текстових повідомлень на природній мові, що дозволило виділити основні етапи при проведенні дослідження текстових повідомлень в системах обміну повідомленнями.

5. Розроблено алгоритм виділення та підрахунку рівня агресії в текстових повідомленнях на основі використання алгоритмів обробки природніх мов, що дозволило спроектувати та проаналізувати структуру програмного додатку аналізу текстових повідомлень.

6. Проведено програмну реалізацію програмного додатку опрацювання текстових повідомлень на основі запропонованого алгоритму оцінки рівня агресії та запропонованої внутрішньої архітектури програмного додатку, що дозволило провести тестування різнотипних текстових повідомлень на визначення рівня агресії.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Тимчук Є.І., Далекий М.Р. Оцінка якості програмного коду на основі аналізу рівня його читабельності. Збірник тез VIII Науково-практична конференція молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі», Тернопіль, 05 грудня 2023 р. с. 45.
2. Далекий М.Р., Тимчук Є.І. Оцінка рівня агресії на основі аналізу текстових повідомлень. Збірник тез VIII Науково-практична конференція молодих вчених і студентів «Інтелектуальні комп'ютерні системи та мережі», Тернопіль, 05 грудня 2023 р. с. 37.
3. Дубчак Л. О., Гураль І. В. Методичні вказівки до оформлення курсових проектів, звітів про проходження практики, випускних кваліфікаційних робіт для студентів спеціальності «Комп'ютерна інженерія» / ред. О. М. Березький. Тернопіль : ТНЕУ, 2019. 33 с.
4. Дубчак Л. О., Мельник Г. М. Методичні рекомендації до виконання кваліфікаційної роботи з освітнього ступеня “Магістр”. Спеціальність: 123 - Комп'ютерна інженерія. Магістерська програма – Комп'ютерна інженерія" / ред. О. М. Березький. Тернопіль : ЗУНУ, 2020. 32 с.
5. Adorjan M., Ricciardelli R. Relational aggression. *Cyber-Risk and Youth*. Abingdon, Oxon ; New York, NY : Routledge, 2019., 2018. P. 70–90. URL: <https://doi.org/10.4324/9781315158686-5> (date of access: 21.11.2023).
6. Antipina S. S., Bakhvalova E. V., Miklyaeva A. V. Gender differences in adolescent cyber-aggression. *The Herzen University Conference on Psychology in Education*. 2019. URL: <https://doi.org/10.33910/herzenpsyconf-2019-2-7> (date of access: 21.11.2023).
7. Antipina S. S. Social intelligence and cyber-aggression in adolescents. *Psychology in Education*. 2022. Vol. 4, no. 2. P. 133–144. URL: <https://doi.org/10.33910/2686-9527-2022-4-2-133-144> (date of access: 21.11.2023).

8. Bilingual Cyber-aggression detection on social media using LSTM autoencoder / K. Kumari et al. *Soft Computing*. 2021. Vol. 25, no. 14. P. 8999–9012. URL: <https://doi.org/10.1007/s00500-021-05817-y> (date of access: 21.11.2023).

9. Corcoran L., Guckin C., Prentice G. Cyberbullying or Cyber Aggression?: A Review of Existing Definitions of Cyber-Based Peer-to-Peer Aggression. *Societies*. 2015. Vol. 5, no. 2. P. 245–255. URL: <https://doi.org/10.3390/soc5020245> (date of access: 21.11.2023).

10. Cyber Bullying: A Critical Overview. *Aggression and Violence*. New York, NY : Routledge, 2016., 2016. P. 235–250. URL: <https://doi.org/10.4324/9781315524696-22> (date of access: 21.11.2023).

11. Cyber-Victimization and Cyber-Aggression among Portuguese Adolescents / M. J. D. Martins et al. *International Journal of Cyber Behavior, Psychology and Learning*. 2016. Vol. 6, no. 3. P. 65–78. URL: <https://doi.org/10.4018/ijcbpl.2016070105> (date of access: 21.11.2023).

12. Cyber-Victimization and Cyber-Aggression / M. J. D. Martins et al. *Analyzing Human Behavior in Cyberspace*. 2019. P. 255–271. URL: <https://doi.org/10.4018/978-1-5225-7128-5.ch014> (date of access: 21.11.2023).

13. Cyber-Victimization and Cyber-Aggression / M. J. D. Martins et al. *Research Anthology on Rehabilitation Practices and Therapy*. 2021. P. 1273–1285. URL: <https://doi.org/10.4018/978-1-7998-3432-8.ch063> (date of access: 21.11.2023).

14. Ghosal S., Jain A. Research Journey of Hate Content Detection From Cyberspace. *Research Anthology on Combating Cyber-Aggression and Online Negativity*. 2022. P. 542–567. URL: <https://doi.org/10.4018/978-1-6684-5594-4.ch031> (date of access: 21.11.2023).

15. Grigg D. W. Cyber-Aggression: Definition and Concept of Cyberbullying. *Australian Journal of Guidance and Counselling*. 2010. Vol. 20, no. 2. P. 143–156. URL: <https://doi.org/10.1375/ajgc.20.2.143> (date of access: 21.11.2023).

16. Hollá K., Sender B., Kosovac S. Empathy in the Prevention of Cyber-aggression. *Journal of Education, Teaching and Social Studies*. 2023. Vol. 5, no. 3. P. p59. URL: <https://doi.org/10.22158/jetss.v5n3p59> (date of access: 21.11.2023).

17. Kumari K., Singh J. P. Multi-modal cyber-aggression detection with feature optimization by firefly algorithm. *Multimedia Systems*. 2021. URL: <https://doi.org/10.1007/s00530-021-00785-7> (date of access: 21.11.2023).
18. Mladenović M., Ošmjanski V., Stanković S. V. Cyber-aggression, Cyberbullying, and Cyber-grooming. *ACM Computing Surveys*. 2021. Vol. 54, no. 1. P. 1–42. URL: <https://doi.org/10.1145/3424246> (date of access: 21.11.2023).
19. Runions K. C., Bak M., Cross D. Cyber Aggression. *Encyclopedia of Adolescence*. Cham, 2016. P. 1–10. URL: https://doi.org/10.1007/978-3-319-32132-5_780-1 (date of access: 21.11.2023).
20. Runions K. C., Bak M., Cross D. Cyber Aggression. *Encyclopedia of Adolescence*. Cham, 2018. P. 834–843. URL: https://doi.org/10.1007/978-3-319-33228-4_780 (date of access: 21.11.2023).
21. Runions K. C., Bak M. Online Moral Disengagement, Cyberbullying, and Cyber-Aggression. *Cyberpsychology, Behavior, and Social Networking*. 2015. Vol. 18, no. 7. P. 400–405. URL: <https://doi.org/10.1089/cyber.2014.0670> (date of access: 21.11.2023).
22. Runions K. C., Bak M., Shaw T. Disentangling functions of online aggression: The Cyber-Aggression Typology Questionnaire (CATQ). *Aggressive Behavior*. 2016. Vol. 43, no. 1. P. 74–84. URL: <https://doi.org/10.1002/ab.21663> (date of access: 21.11.2023).
23. Wong-Lo M., Bullock L. M., Gable R. A. Cyber bullying: practices to face digital aggression. *Emotional and Behavioural Difficulties*. 2011. Vol. 16, no. 3. P. 317–325. URL: <https://doi.org/10.1080/13632752.2011.595098> (date of access: 21.11.2023).
24. Yinka-Banjo C., Raji G. L., Ohaleté I. P. Auto-Detection of Human Factor Contents on Social Media Posts Using Word2vec and Long Short-Term Memory (LSTM). *Research Anthology on Combating Cyber-Aggression and Online Negativity*. 2022. P. 160–171. URL: <https://doi.org/10.4018/978-1-6684-5594-4.ch010> (date of access: 21.11.2023).

25. Andersson V. Machine Learning in Logistics: Machine Learning Algorithms : Data Preprocessing and Machine Learning Algorithms : thesis. 2017. URL: <http://urn.kb.se/resolve?urn=urn:nbn:se:ltu:diva-64721> (date of access: 21.11.2023).
26. Ayyadevara V. K. Artificial Neural Network. Pro Machine Learning Algorithms. Berkeley, CA, 2018. P. 135–165. URL: https://doi.org/10.1007/978-1-4842-3564-5_7 (date of access: 21.11.2023).
27. Ayyadevara V. K. Basics of Machine Learning. Pro Machine Learning Algorithms. Berkeley, CA, 2018. P. 1–15. URL: https://doi.org/10.1007/978-1-4842-3564-5_1 (date of access: 21.11.2023).
28. Ayyadevara V. K. Pro Machine Learning Algorithms. Berkeley, CA : Apress, 2018. URL: <https://doi.org/10.1007/978-1-4842-3564-5> (date of access: 21.11.2023).
29. Fernandes de Mello R., Antonelli Ponti M. Assessing Supervised Learning Algorithms. Machine Learning. Cham, 2018. P. 129–161. URL: https://doi.org/10.1007/978-3-319-94989-5_3 (date of access: 21.11.2023).
30. Geetha T. V., Sendhilkumar S. Classification Algorithms. Machine Learning. Boca Raton, 2023. P. 127–151. URL: <https://doi.org/10.1201/9781003290100-6> (date of access: 21.11.2023).
31. Golden R. M. Formal Machine Learning Algorithms. Statistical Machine Learning. First edition. j Boca Raton, FL : CRC Press, 2020. j Includes bibliographical references and index., 2020. P. 83–96. URL: <https://doi.org/10.1201/9781351051507-3> (date of access: 21.11.2023).
32. Li F., Lai L., Cui S. Machine Learning Algorithms. Cham : Springer International Publishing, 2022. URL: <https://doi.org/10.1007/978-3-031-16375-3> (date of access: 21.11.2023).
33. Li F., Lai L., Cui S. Summary and Extensions. Machine Learning Algorithms. Cham, 2022. P. 83–85. URL: https://doi.org/10.1007/978-3-031-16375-3_5 (date of access: 21.11.2023).
34. Ling Q. Machine learning algorithms review. Applied and Computational Engineering. 2023. Vol. 4, no. 1. P. 91–98. URL: <https://doi.org/10.54254/2755-2721/4/20230355> (date of access: 21.11.2023).

35.Mitra M. Algorithms and Machine Learning. American Research Journal of Electronics and Communication Engineering. 2019. Vol. 1, no. 1. URL: <https://doi.org/10.21694/2643-3486.19008> (date of access: 21.11.2023).

36.Schwalbe U. Algorithms, Machine Learning, and Collusion. SSRN Electronic Journal. 2018. URL: <https://doi.org/10.2139/ssrn.3232631> (date of access: 21.11.2023).

37.Chowdhury G. G. Natural language processing. Annual Review of Information Science and Technology. 2005. Vol. 37, no. 1. P. 51–89. URL: <https://doi.org/10.1002/aris.1440370103> (date of access: 21.11.2023).

38.Covington M. A., Pereira F. C. N., Grosz B. J. Natural Language Processing. Language. 1995. Vol. 71, no. 3. P. 652. URL: <https://doi.org/10.2307/416262> (date of access: 21.11.2023).

39.Hamandi A. M., Bahjat H., Abdul Hassan A. K. Natural Language Processing Using Natural Language Toolkit. المعلومات لتكنولوجيا العراقية المجلة. 2016. P. 70. URL: <https://doi.org/10.34279/0923-007-002-010> (date of access: 21.11.2023).

40.HASHIDA K. Natural Language Processing. The Journal of the Institute of Electrical Engineers of Japan. 2001. Vol. 121, no. 3. P. 195–198. URL: <https://doi.org/10.1541/ieejjournal.121.195> (date of access: 21.11.2023).

41.Jain A., Kulkarni G., Shah V. Natural Language Processing. International Journal of Computer Sciences and Engineering. 2018. Vol. 6, no. 1. P. 161–167. URL: <https://doi.org/10.26438/ijcse/v6i1.161167> (date of access: 21.11.2023).

42.Németh R., Koltai J. Natural language processing. Intersections. 2023. Vol. 9, no. 1. P. 5–22. URL: <https://doi.org/10.17356/ieejsp.v9i1.871> (date of access: 21.11.2023).

43.Niimi A. Natural Language Processing. Chest. 2021. Vol. 159, no. 6. P. 2149–2150. URL: <https://doi.org/10.1016/j.chest.2021.01.045> (date of access: 21.11.2023).

44.N. O. Sadiku M., Zhou Y., M. Musa S. Natural Language Processing. International Journal of Advances in Scientific Research and Engineering. 2018. Vol.

4, no. 5. P. 68–70. URL: <https://doi.org/10.31695/ijasre.2018.32708> (date of access: 21.11.2023).

45. Patten T., Jacobs P. Natural-language processing. *IEEE Expert*. 1994. Vol. 9, no. 1. P. 35. URL: <https://doi.org/10.1109/64.295134> (date of access: 21.11.2023).

46. Selfridge M. Natural language processing. *Artificial Intelligence in Engineering*. 1987. Vol. 2, no. 1. P. 50. URL: [https://doi.org/10.1016/0954-1810\(87\)90076-8](https://doi.org/10.1016/0954-1810(87)90076-8) (date of access: 21.11.2023).

47. V R., Naik S. NATURAL LANGUAGE PROCESSING. *International Journal of Computing Algorithm*. 2023. Vol. 12, no. 2. URL: <https://doi.org/10.20894/ijcoa.101.012.002.001> (date of access: 21.11.2023).

48. Wilks Y. Natural language processing. *Communications of the ACM*. 1996. Vol. 39, no. 1. P. 60–62. URL: <https://doi.org/10.1145/234173.234180> (date of access: 21.11.2023).

49. Yilmaz A. E. Natural Language Processing. *International Journal of Systems and Service-Oriented Engineering*. 2014. Vol. 4, no. 1. P. 68–83. URL: <https://doi.org/10.4018/ijssoe.2014010105> (date of access: 21.11.2023).

50. Литвинчук Л. ТОЛЕРАНТНІСТЬ ТА АГРЕСІЯ ЯК БІПОЛЯРНІ СКЛАДОВІ СПІЛКУВАННЯ В ІНТЕРНЕТІ. *Перспективи та інновації науки*. 2022. № 10(15). URL: [https://doi.org/10.52058/2786-4952-2022-10\(15\)-486-494](https://doi.org/10.52058/2786-4952-2022-10(15)-486-494) (дата звернення: 21.11.2023).