НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ АТОМНОЇ ТА ТЕПЛОВОЇ ЕНЕРГЕТИКИ

кафедра ЦИФРОВИХ ТЕХНОЛОГІЙ В ЕНЕРГЕТИЦІ

"На правах рукопису"
УДК_____

“До захисту допущено”
Завідувач кафедри ЦТЕ
_____ Наталія АУШЕВА
“___”_____2023 р.

Магістерська дисертація

на здобуття ступеня магістра
за освітньо-професійною програмою
“Цифрові технології в енергетиці”
зі спеціальності *122 «Комп’ютерні науки»*

на тему**:** “Визначення дезінформації в текстах засобів масової інформації та соціальних мереж”

Виконав: студент 2 курсу, групи ТР-22мп

Головакін Микита Андрійович _____
(прізвище, ім'я та по батькові) (підпис)

Науковий керівник: доцент, к.т.н., Світлана ШАПОВАЛОВА _____
(посада, науковий ступінь, вчене звання, прізвище, ім'я та по батькові) (підпис)

Рецензент: доцент кафедри ІПЗЕ, к.т.н. Валерій КУЗЬМІНИХ _____
(посада, науковий ступінь, вчене звання, прізвище, ім'я та по батькові) (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2023

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ АТОМНОЇ ТА ТЕПЛОВОЇ ЕНЕРГЕТИКИ

Кафедра ЦИФРОВИХ ТЕХНОЛОГІЙ В ЕНЕРГЕТИЦІ

Рівень вищої освіти – другий (магістерський)

Спеціальність 122 “Комп’ютерні науки”

Освітньо-професійна програма “Цифрові технології в енергетиці”

ЗАТВЕРДЖУЮ
Завідувач кафедри ЦТЕ
_____ **Наталія АУШЕВА**
“____” _____ 2023 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Головакіну Микиті Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема дисертації “Визначення дезінформації в текстах засобів масової інформації та соціальних мереж”

Науковий керівник дисертації: ***Шаповалова Світлана Ігорівна, к.т.н., доцент***
(прізвище, ім’я, по батькові науковий ступінь, вчене звання)

затверджені наказом по університету від “06” листопада 2023р №5152-с

2. Термін подання студентом дисертації 18 грудня 2023р.

3. Об'єкт дослідження: методи та моделі штучного інтелекту для класифікації тексту природньої мови.

4. Вихідні дані до роботи: Тексти природньою мовою засобів масової інформації та дописів соціальних мереж англійською мовою.

5. Перелік завдань, які потрібно розробити: Провести аналіз підходів, для векторизації та класифікації текстів для визначення дезінформації у ЗМІ та

дописах природньої мови, представити обрані методи, зібрати тестову вибірку даних, представити обчислювальний експеримент з вибором найоптимальнішої моделі та провести аналіз стартап-проєкту.

6. Орієнтовний перелік ілюстративного матеріалу: Архітектура SBERT, блок-схема методу голосування більшості голосів, результати досліджень.

7. Орієнтовний перелік публікацій: Detecting disinformation in english-language posts on social media. *Technical sciences Innovative scientific research: theory and practice: Proceedings of the X International Scientific and Practical Conference*. November 21-24, 2023. Stockholm, Sweden.

8. Дата видачі завдання « 24 » жовтня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Строки виконання етапів магістерської дисертації	Примітка
1	Аналіз методів, підходів та програмних засобів	24.10.2023 – 29.10.2023	
2	Представлення методів векторизації та класифікації текстів природньої мови	30.10.2023 – 03.11.2023	
3	Проведення обчислювальних експериментів для вибору кращих моделей	06.11.2023 – 17.11.2023	
4	Розробка системи	20.11.2023 – 1.12.2023	
5	Оформлення магістерської дисертації	04.12.2023 – 11.12.2023	
6	Захист	08.01.2024 – 20.01.2024	

Студент _____ **Микита ГОЛОВАКІН**
(підпис) (ім'я, ПРІЗВИЩЕ)

Науковий керівник _____ **Світлана ШАПОВАЛОВА**
(підпис) (ім'я, ПРІЗВИЩЕ)

АНОТАЦІЯ

Актуальність роботи:

Дослідження, спрямовані на виявлення дезінформації, що міститься в текстах, які поширюються ЗМІ та соціальними мережами у воєнний час, набувають все більшої актуальності та стратегічної важливості. У цьому контексті дослідження та розробка методів виявлення та протидії дезінформації у воєнний час має стратегічне значення.

Тому наукові дослідження з виявлення дизінформації у воєнний час є актуальними, а науковці, експерти з інформаційної безпеки та політики повинні зосередити свої зусилля на розробці та вдосконаленні методів виявлення та запобігання дезінформації в таких ситуаціях.

Мета:

Визначити оптимальні за точністю методи класифікації дезінформації в текстах, опублікуваних у засобах масової інформації та соціальних мережах.

Завдання роботи:

1. Провести дослідження з визначення методів та наявного програмного забезпечення виявлення дизінформації у повідомленнях.
2. Представити методи векторизації та класифікації тексту у моделях штучного інтелекту.
3. Сформувати датасет та реалізувати методи виявлення дизінформації в текстових повідомленнях.
4. Провести обчислювальні експерименти.
5. На основі проведених досліджень та розробок представити стартап-стратегію.

Об'єкт дослідження:

методи та моделі штучного інтелекту для класифікації тексту природньої мови.

Предмет дослідження:

методи та моделі штучного інтелекту для виявлення дизінформації в текстах природньої мови.

Фінальним результатом досліджень буде створена система, що володіє високою функціональністю. Вхідним потоком для системи може бути або речення, або текст з інформацією. Користувач самостійно визначає, чи є це дизінформацією чи ні, за допомогою зручного веб-інтерфейсу. Такий підхід надає користувачеві чітку інтерпретацію результатів та дозволяє вчасно виявляти небезпечні джерела інформації.

Апробація результатів дисертації:

Основні положення даної роботи обговорювались на Technical sciences Innovative scientific research: theory and practice: Proceedings of the X International Scientific and Practical Conference у 21-24 листопада, 2023, м. Стокгольм, Швеція.

Структура та обсяг роботи: Магістерська дисертація складається з вступу, пяти розділів та висновків.

Ключові слова: дезінформація, NLP, SBERT, XLNet.

ABSTRACT

Relevance of the work:

Research aimed at detecting disinformation contained in texts disseminated by the media and social networks in wartime is becoming increasingly relevant and strategically important. In this context, research and development of methods for detecting and countering disinformation in wartime is of strategic importance.

Therefore, scientific research on detecting disinformation in wartime is relevant, and scientists, information security experts and policy makers should focus their efforts on developing and improving methods to detect and prevent disinformation in such situations.

Objective:

To determine the best methods for classifying disinformation in texts published in the media and social networks.

Tasks:

1. Conduct research to identify methods and available software for detecting disinformation in messages.
2. Present methods of text vectorisation and classification in artificial intelligence models.
3. Create a dataset and implement methods for detecting disinformation in text messages.
4. Conduct computational experiments.
5. Based on the research and development, present a startup strategy.

Object of research:

Artificial intelligence methods and models for natural language text classification.

Subject of research:

Artificial intelligence methods and models for detecting disinformation in natural language texts.

The final result of the research will be a system with high functionality. The input stream for the system can be either a sentence or a text with information. The user determines whether it is disinformation or not using a user-friendly web interface. This approach provides the user with a clear interpretation of the results and allows them to identify dangerous sources of information in time.

Testing the results of the thesis:

The main provisions of this work were discussed at Technical sciences Innovative scientific research: theory and practice: Proceedings of the X International Scientific and Practical Conference on 21-24 November, 2023, Stockholm, Sweden.

Structure and scope of work: The master's thesis consists of an introduction, five chapters and conclusions.

Keywords: disinformation, NLP, SBERT, XLNet.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ 10
ВСТУП 11
1 АНАЛІЗ ДЕЗІНФОРМАЦІЇ В КОНТЕКСТІ ВОЄННОГО СТАНУ 12
1.1 Методи досліджень 12
1.1.1 Методи векторизації тексту 12
1.1.2 Методи класифікації для аналізу тексту 16
1.2 Датасети 19
1.3 Огляд існуючих рішень 20
1.3.1 Сервіс FactCheck.org 20
1.3.2 Сервіс Snopes 22
1.3.3 Розширення NewsGuard 23
Висновки до розділу 1 23
2 МЕТОДИ ВЕКТОРИЗАЦІЇ ТА РОЗПІЗНАВАННЯ ДЕЗІНФОРМАЦІЇ В ТЕКСТОВИХ ПОВІДОМЛЕННЯХ 24
2.1 Передобробка тексту за допомогою RAKE 24
2.2 Векторизація тексту у моделі SBERT 25
2.3 Методи класифікації у моделі SBERT 28
2.4 Векторизація тексту у моделі XLNet Large 31
2.5 Методи класифікації у моделі XLNet Large 33
2.6 Прийняття рішення класифікації тексту на наявність дезінформації 35
2.7 Тренування моделі SBERT 37
2.8 Тренування моделі XLNet Large 39
Висновки до розділу 2 40
3 ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДУ ВИЯВЛЕННЯ ДЕЗІНФОРМАЦІЇ В ТЕКСТАХ ЗМІ 41
3.1 Вибір технологій для реалізації бекенду 41

3.1.1 Мова програмування Python 42
3.1.2 Фреймворк інтеграції з фронтенд-частиною FastAPI 44
3.1.3 Бібліотеки для попередньої обробки тексту природньої мови 44
3.1.4 Бібліотека для семантичної обробки текстів природньої мови SBERT 45
3.1.5 Бібліотека для семантичної обробки текстів природньої мови XLNet Large 45
3.2 Вибір технологій для реалізації фронтенду 46
3.2.1 Мова програмування JavaScript 46
3.2.2 Фреймворк створення інтерфейсів React 47
3.2.3 Бібліотеки реалізації дизайну та стилізації Bootstrap 48
3.2.4 Бібліотеки для візуалізації даних 49
3.3 Збір та індексація даних 50
3.3.1 Пошуковий двигун ElasticSearch 50
3.3.2 Збір еталонних текстів природньої мови у датасет 51
3.3.3 Збір текстів дописів природньої мови у датасет 52
3.4 Реалізація бекенд-частини програмного продукту 53
3.4.1 Реалізація передобробки тексту 54
3.4.2 Реалізація FastAPI 55
3.4.3 Реалізація ступеню визначення дизінформації у Python 56
3.4.4 Налаштування індексації у ElasticSearch 57
3.5 Реалізація фронтенд-частини програмного продукту 58
3.5.1 Інтерфейс користувача при вході у систему 60
3.5.2 Відображення графіків датасету програмного продукту 61
3.5.3 Введення даних в текстове поле програмного продукту 62
3.5.4 Введення даних через файл у інтерфейсі програмного продукту 65
3.5.5 Відображення ключових слів у інтерфейсі користувача 68
3.5.6 Підсумок роботи аналітика по виявленню дезінформацію 69
Висновки до розділу 3 70

4 ОБЧИСЛЮВАЛЬНІ ЕКСПЕРИМЕНТИ МОДЕЛЕЙ ОБРОБКИ ПРИРОДНЬОГО ТЕКСТУ 72
4.1 Порівняння точності моделей схожості тексту сімейства BERT 72
4.1.1 Оцінка якості класифікації моделі RoBERTa-base 73
4.1.2 Оцінка якості класифікації моделі All-MiniLM-L6-v2 75
4.1.3 Оцінка якості класифікації моделі all-mpnet-base-v2 76
4.2 Аналіз результатів та вибір моделі порівняння схожості тексту 77
4.3 Порівняння точності моделей класифікації тексту 78
4.3.1 Оцінка якості класифікації моделі xlnet-large-cased 78
4.3.2 Модель bert-base-uncased 80
4.4 Аналіз результатів та вибір моделі класифікації тексту 80
4.5 Архітектура роботи ансамблю моделей порівняння схожості тексту та класифікації 81
Висновки до розділу 4 83
5 РОЗРОБКА СТАРТАП-ПРОЄКТУ 84
5.1 Опис ідеї 84
5.2 Технологічний аудит проекту 85
5.3 Аналіз ринкових можливостей 86
5.4 Розробка ринкової стратегії стартап-проекту 93
5.5 Розроблення маркетингової програми стартап проекту 95
Висновки до розділу 5 96
ВИСНОВКИ 97
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ 98
Додаток А 102

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

ЗМІ – Засоби масової інформації.

SBERT – модель Sentence-BERT.

Датасет – Набір даних упорядкованих та зібраних певним чином.

GloVe – Global Vectors for Word Representation.

NLTK – Natural Language Toolkit.

LSA - Latent Semantic Analysis.

ВСТУП

Воєнний час створює ідеальне середовище для поширення дезінформації. Державні актори, недержавні суб'єкти та терористичні організації можуть використовувати цей засіб для досягнення різноманітних цілей. Дезінформація може бути спрямована не лише на населення, а й на міжнародних партнерів та міжнародну спільноту. Це може призвести до маніпулювання громадською думкою, соціальної дестабілізації, втручання в політичні процеси і навіть до інтенсифікації конфліктів [1].

Поширення дезінформації у воєнний час також може мати серйозні геополітичні наслідки, такі як зміна динаміки міжнародних відносин і загроза міжнародній безпеці. Наприклад, інформаційні кампанії можуть впливати на підтримку міжнародних альянсів і ставлення інших держав до конфлікту [1].

У цьому контексті дослідження та розробка методів виявлення та протидії дезінформації у воєнний час має стратегічне значення. Вони можуть допомогти захистити інформаційну безпеку, підвищити обізнаність громадськості та вдосконалити методи захисту від інформаційних маніпуляцій. Розробка таких методів не лише сприяє забезпеченню стабільності та безпеки під час війни та конфліктів, але й підвищує ефективність роботи міжнародної спільноти.

Тому актуальність наукових досліджень на цю тему у воєнний час є важливою, а науковці, експерти з інформаційної безпеки та політики повинні зосередити свої зусилля на розробці та вдосконаленні методів виявлення та запобігання дезінформації в таких ситуаціях.

1 АНАЛІЗ ДЕЗІНФОРМАЦІЇ В КОНТЕКСТІ ВОЄННОГО СТАНУ

У першому розділі визначено актуальність, мету, задачу і постановку завдань та методику досліджень, спрямованих на виявлення дизінформації у текстах та повідомленнях. Також розглянуті наявні ресурси, призначені для виявлення дизінформації в умовах воєнного конфлікту.

1.1 Методи досліджень

Задачею дослідження є вирішення двох ключових завдань: перше – формалізація природного тексту з метою створення чіткої структури та аналізу; друге – виявлення дезінформації в текстових джерелах, що визначається на підставі введених користувачем маркерів та оцінок, забезпечуючи точність та об'єктивність процесу аналізу.

1.1.1 Методи векторизації тексту

На розгляд дано основні методи векторизації тексту.

Bag-of-Words (Мішок слів) є одним з основних методів векторизації тексту у сфері обробки природної мови [2]. Зародився цей метод в ранній історії обчислювальної лінгвістики та обробки текстової інформації. Основна мета виникнення методу була спростування складнощів роботи із семантичними аспектами текстів та створення ефективної системи обробки текстових даних. Метод мішка слів ґрунтується на ідеї розглядання тексту як набору слів без врахування їх порядку чи семантики. Кожен документ представляється у вигляді

вектора, де кожна координата відповідає окремому слову, а значення цієї координати вказує на кількість входжень відповідного слова у текст. Такий підхід забезпечує простоту та ефективність у роботі з текстовими даними, але втрачає важливість синтаксичних та семантичних зв'язків у тексті.

Метод TF-IDF (Term Frequency-Inverse Document Frequency – Частота Терміна-Обернена Частота Документа) виник у зв'язку із потребами інформаційного пошуку та обробки текстів [3]. Розроблений Карен Спарк Джонсоном у 1957 році та пізніше вдосконалений Джорджем Салтоном та Сіріусом Тайджо, цей метод призначений для врахування не лише частоти слова у конкретному документі, а й його значущості у загальному контексті. Ідея TF-IDF полягає в тому, щоб визначити вагу кожного слова відповідно до його частоти у документі та обертання на частоту вживання у всьому корпусі текстів. Це дозволяє виділити ключові терміни, які характеризують конкретний документ в контексті усього корпусу.

Метод векторизації слів, відомий як Word Embeddings [4], є продуктом розвитку глибокого навчання та нейромережевих технологій. Цей метод став особливо популярним у середині 2010-х років завдяки своїй здатності представляти слова у векторному просторі, зберігаючи семантичні зв'язки між ними. Word Embeddings використовує нейромережеві архітектури для створення векторів слів, де кожен вектор представляє семантичні властивості відповідного слова. Розглянемо дві основні моделі векторизації слів: Word2Vec та GloVe. Word2Vec, розроблений Томасом Міколовим та його групою у Google, використовує нейромережу для оцінки ймовірності сусідніх слів за умови конкретного слова. Модель може бути навчена з великим корпусом текстів, де кожне слово представляється у векторному просторі. Так, слова зі схожими значеннями матимуть схожі вектори. GloVe (Global Vectors for Word Representation), розроблений станом на 2014 рік, використовує глобальні статистики спільного входження слів у корпусі. Модель розглядає частоту спільного входження двох слів, щоб визначити їхню взаємну асоціацію. Знову ж

таки, слова, які часто зустрічаються разом, будуть мати схожі вектори. Word Embeddings дозволяє захопити семантичні зв'язки між словами, враховуючи їхнє контекстове вживання. Наприклад, вектор для слова “кіт” може вказати його близькість до інших слів, таких як “тварина”, “домашній”, що відображає його семантичне значення.

N-грамні моделі становлять підхід до обробки природної мови, який акцентує увагу на послідовностях слів фіксованої довжини, відомих як n-грами [5]. Виникши в контексті розвитку статистичної обробки мови та машинного навчання, цей метод спрямований на врахування контексту та взаємозв'язків між словами. Модель N-грам визначає ймовірність появи кожної N-грами в тексті, використовуючи умовні ймовірності, де ймовірність наступного слова залежить від попередніх n-1 слів. Наприклад, у біграмній моделі речення “Кіт сидить на дивані” розглядається як два блоки: {“Кіт”, “сидить”}, {“сидить”, “на”}, {“на”, “дивані”}. Зазначені моделі використовуються для прогнозування слів у тексті, виявлення зв'язків між ними та побудови ймовірнісних моделей для автоматичної обробки текстової інформації. Вони є корисним інструментом для аналізу мови, генерації тексту та розв'язання завдань, пов'язаних із структурою та контекстом слів.

Latent Semantic Analysis (латентний семантичний аналіз) є методом обробки природної мови та векторизації тексту, який спрямований на виявлення латентних (прихованих) семантичних зв'язків між словами у тексті [6]. Розроблений у галузі інформаційного пошуку та обробки текстової інформації, LSA використовується для аналізу та розуміння схожості між документами та термінами. Метод LSA базується на техніці сингулярного розкладання матриці, що представляє собою матричний розклад з врахуванням особливостей. Для створення матриці, кожен рядок якої відповідає слову, а кожен стовпець – документу, використовується термін-документна матриця, де кожен елемент вказує на частоту зустрічі терміну у документі. Латентний семантичний аналіз зменшує розмірність цієї матриці, утримуючи тільки найбільш інформативні особливості. Це дозволяє моделі

враховувати схожість між словами та документами навіть у випадках, коли вони не взаємодіють напряму. LSA використовується для тематичного моделювання, кластеризації документів та покращення завдань інформаційного пошуку. Цей метод робить можливим виявлення прихованих семантичних зв'язків у тексті, що поліпшує розуміння змісту та забезпечує кращі результати у задачах обробки природної мови.

BERT (Bidirectional Encoder Representations from Transformers) є сучасною та впливовою моделлю для векторизації тексту, основаною на архітектурі трансформерів [7]. Розроблений у 2018 році компанією Google, BERT відзначається здатністю розуміти семантику слів та фраз в контексті їхнього оточення. BERT використовує трансформатори, щоб ефективно моделювати взаємодії між словами в тексті. Однак його ключовою особливістю є бідирекційність: він враховує інформацію з обох сторін кожного слова в контексті. Це дозволяє моделі краще розуміти семантику слів у реальних реченнях. BERT генерує вектори (ембедінги) для слів та фраз, що відображають їхню семантичну інформацію. Ці вектори можуть бути використані для вирішення різних завдань обробки природної мови, таких як класифікація тексту, витягування інформації, аналіз тональності тощо. BERT Embeddings отримали визнання завдяки своїй здатності вдачно враховувати контекст та залежності між словами, що дозволяє моделі отримувати багатоінформативні представлення тексту. Ця модель стала основою для багатьох передових застосувань у сфері обробки природної мови та вирішення широкого спектру завдань.

Оскільки необхідно побудувати найсучаснішу систему розпізнавання дизінформації, яка має підтекст, та необхідно враховувати залежності між словами та реченнями, модель BERT найкраще підходить для розв’язання цієї задачі.

1.1.2 Методи класифікації для аналізу тексту

На розгляд дано найпоширеніші методи класифікації для аналізу тексту.

Naive Bayes classifier (наївні байєсівські класифікатори) засновані на теорії ймовірностей і правилі Байєса і з'явилися в контексті розвитку статистичної обробки мови [8]. Вони називаються “наївними”, оскільки припускають абсолютну незалежність між ознаками, що робить їх простим і ефективним методом класифікації текстових даних. Ідея застосування теорії ймовірностей до класифікації текстів була вперше сформульована у 18 столітті англійським математиком Томасом Байєсом. Однак саме припущення наївності, коли всі ознаки вважаються незалежними, визначає основний принцип цього класифікатора. Наївні байєсівські класифікатори широко використовуються не тільки в обробці природної мови, але й в інших галузях, таких як фінанси та медицина. Їх широке використання пояснюється поєднанням простоти і хороших практичних результатів. Хоча вони не можуть обробляти складні текстові структури, їх часто достатньо для вирішення багатьох завдань класифікації текстів.

Support Vector Machines (Метод опорних векторів) – це потужні алгоритми машинного навчання, що використовуються для класифікації та регресії [9]. Метод розвинувся зі статистики та теорії оптимізації, і його основна ідея полягає в тому, щоб знайти найкращу гіперплощину для розділення класів у просторі ознак. Ця гіперплощина вибирається так, щоб максимізувати відстань до найближчого екземпляра кожного класу (так званий “опорний вектор”). Методи опорних векторів можна використовувати для вирішення як бінарних, так і багатокласових задач класифікації. Методи опорних векторів також ефективні при роботі з великими обсягами даних, є універсальними і добре працюють з новими і невідомими даними. Вперше методи опорних векторів були представлені Вапніком і Червоненкісом у 1963 році, але розвинулися між 1990 і 2000 роками з появою нових методів оптимізації і зростанням нейронних мереж.

Сьогодні SVM широко використовуються в різних галузях, таких як обробка тексту, комп'ютерний зір, фінанси, медицина та біоінформатика.

Deep Learning (Глибоке навчання) – це передовий підхід до машинного навчання, заснований на використанні глибоких нейронних мереж [10]. Цей підхід з'явився в результаті збільшення обчислювальних потужностей і розвитку алгоритмів навчання. Глибинні нейронні мережі використовують багатошарову архітектуру і можуть автоматично вивчати уявлення та особливості з даних. Однією з основних архітектур глибокого навчання є згорткова нейронна мережа (CNN), яка спеціалізується на обробці великих обсягів візуальних даних, таких як зображення та відео. Іншою важливою архітектурою є рекурентні нейронні мережі (RNN), які ефективно працюють з послідовними даними, такими як текст і часові ряди.

Ensemble learning (ансамблеві методи) – це потужний та ефективний підхід у машинному навчанні, який об'єднує результати кількох базових моделей для отримання більш точних та стабільних прогнозів [11]. Цей підхід зменшує ризик перенавчання, підвищує точність і надійність прогнозів та покращує загальну продуктивність моделі. Однією з основних ансамблевих стратегій є багатокласова класифікація, коли кілька моделей роблять прогнози, а їхні результати об'єднуються, часто з використанням голосування та усереднення.

Ансамблі можуть бути сформовані за допомогою різних методів, зокрема:

Bootstrap-агрегація передбачає навчання декількох екземплярів одного алгоритму на різних підвибірках даних і об'єднання їхніх прогнозів. Це зменшує ризик перенавчання та покращує стабільність моделі.

Методи Boosting (бустінгу) навчають послідовні моделі, де кожна модель виправляє помилки попередньої. Ваги неправильно класифікованих вибірок даних збільшуються, щоб допомогти зменшити помилки [11].

Stacking (стекінг) використовується для об'єднання прогнозів декількох моделей за допомогою метамоделі. Оскільки різні базові моделі можуть

спеціалізуватися на різних аспектах даних, стекування дозволяє їм ефективно поєднувати свої сильні сторони [11].

Random Forest (випадкові ліси) використовують метод пакетування для навчання декількох дерев рішень. Кожне дерево рішень навчається на випадковій підмножині даних, а їхні прогнози об'єднуються [12].

Gradient Boosting (градієнтний бустінг) використовує бустінг для покращення прогнозів шляхом навчання моделей на важливих підмножинах даних відповідно до їхньої важливості [13]. Ансамблеві методи поєднують сильні сторони різних моделей і зменшують їхні спільні недоліки і є важливим компонентом успішних застосувань у різних галузях, включаючи класифікацію, регресію і кластеризацію.

Rule based classification (класифікація на основі правил) – це підхід до машинного навчання, який будує моделі на основі набору логічних правил для автоматичного прийняття рішень про класифікацію об'єктів [14]. У цьому підході створюється певний набір правил, які визначають, які ознаки є важливими для віднесення об'єкта до певного класу. Процес навчання моделі, заснованої на правилах, передбачає визначення та оптимізацію логічних формул, які керують класифікацією. Сюди входять вагові коефіцієнти, порогові значення та умови, які визначають перехід від одного правила до іншого. Після навчання модель може автоматично класифікувати нові об'єкти відповідно до заздалегідь визначених правил. Цей метод легко інтерпретувати, оскільки правила можуть бути перевірені і зрозумілі людиною. Однак він може бути не дуже ефективним у складних завданнях, де важко формалізувати всі можливі правила або де велика кількість ознак взаємодіє одна з одною. Цей метод широко використовується в тих сферах, де важливо, щоб рішення були зрозумілими і легко пояснювалися.

Використання глибинного навчання для виявлення дезінформації є приорітетом, оскільки ця технологія дозволяє моделям автоматично вивчати контекст, семантичні відносини та патерни в тексті. Моделі глибинного навчання, засновані на нейронних мережах, можуть ефективно аналізувати великі обсяги

текстової інформації, виявляти синтаксичні та семантичні аномалії, а також розуміти контекст у висловлюваннях. Ця підхід є потужним інструментом для виявлення маніпуляцій та дезінформації в текстах. Зазначимо, що це саме NLP, оскільки глибинне навчання є частиною області обробки природної мови, яка фокусується на розумінні та обробці мовленнєвої інформації. Було обрано нейромережу XLNet, оскільки вона має найбільшу точність серед існуючих авторегресійних методів.

1.2 Датасети

Наявність набору даних, який можна використовувати для навчання та оцінки моделей ШІ, важлива з кількох причин. По-перше, наявність репрезентативного і ретельно зібраного набору даних є основою для ефективного навчання моделі. Це дозволяє моделі “вивчати” закономірності та взаємозв'язки в наборі даних і робити відповідні розпізнавання та прогнози. По-друге, якість і різноманітність даних впливає на загальну продуктивність моделі. Репрезентативні набори даних дозволяють уникнути упереджень і сприяють створенню моделей, які можуть адекватно реагувати на різні сценарії. Потреба у високоякісних наборах даних визначається їхньою здатністю відображати реальні умови та деталі завдань, які має вирішувати модель. Такі набори даних є важливим кроком на шляху до створення надійних та ефективних моделей штучного інтелекту.

Проаналізуємо існуючі датасети. Для аналізу використовувалась платформа Kaggle [15], яка має безліч датасетів для будь-яких задач. За ключовими словами “Disinformation” було знайдено 1 датасет під назвою “Disinformation and Fake News” [16]. Цей набір даних містить випадки дезінформації, зібрані проектом EuvsDisinfo [17]. Проєкт стартував у 2015 році і займається виявленням,

узагальненням і викриттям випадків дезінформації, що походять з прокремлівських ЗМІ, які поширюються в ЄС і країнах Східного партнерства. Їх база даних містить понад 7000 випадків дезінформації та спростувань. Дані були отримані з API EUvsDisinfo 7 березня 2020 року, а всі веб-сторінки були вилучені 8 березня 2020 року.

Оскільки необхідно вирішити проблему виявлення дизінформації у воєнний час, датасет є не актуальним на час 24 лютого 2022 року. Тому необхідно зібрати датасет про події від початку повномасштабного російського вторгнення в Україну. Таким чином ще одним завданням досліджень є збір даних для нашого датасету.

Для початку було обрано джерела правди журнали The Guardian [18] та The New York Times [19], які містять інформацію про повномасштабне вторгнення. Після обрано джерела інформації про війну у таких джерелах як X та Reddit з хештегами *#WarInUkraine, #StopWarInUkraine* тощо. Вони містять інформацію за період з 24 лютого 2022 року до 15 березня 2022 року. База містить понад 750000 дописів користувачів, які будуть досліджуватись на наявність дизінформації.

Важливо уточнити, що користувач має змогу самому обирати джерела правди та створювати власну базу даних, з якою будуть порівнюватись твердження, які він буде надавати на вхід веб-застосунку. Також від цього залежить точність визначення дезінформації.

1.3 Огляд існуючих рішень

1.3.1 Сервіс FactCheck.org

FactCheck.org – це незалежний інтернет-ресурс, який спеціалізується на факт-чекінгу політичних заяв, стверджень і новин [20]. Цей ресурс був заснований у 2003 році і належить організації Annenberg Public Policy Center при Університеті Пенсильванії. FactCheck.org відомий своєю незалежністю та

об'єктивністю в оцінці інформації, а також своєю детальною роботою з факт-чекінгу.

Діяльність FactCheck.org включає в себе наступні основні елементи:

Факт-чекінг політичних заяв: Однією з основних функцій FactCheck.org є перевірка правильності та обґрунтованості заяв політиків, кандидатів, публічних діячів і різних інших осіб, що мають вплив на політичний дискурс. Це дозволяє громадськості отримувати об'єктивну оцінку інформації, яку надають політичні лідери.

FactCheck.org аналізує новини і статті, які містять сумнівну або спірну інформацію, і намагається визначити, наскільки вони правдиві. Вони досліджують ствердження, представлені у статтях, і надають докладні пояснення їхньої правильності чи неправильності [20].

Результати факт-чекінгу FactCheck.org публікуються на їхньому веб-сайті та інших платформах. Це дозволяє громадськості легко отримувати доступ до об'єктивної інформації щодо стверджень та новин.

Переваги:

FactCheck.org відомий своєю незалежністю від політичних та комерційних впливів, що дозволяє йому надавати об'єктивні результати факт-чекінгу.

Інтернет-ресурс докладно розглядає ствердження та новини, надаючи докази та аргументи, які підтримують їхні висновки.

Результати факт-чекінгу публікуються відкрито на веб-сайті, що робить їх доступними для всіх користувачів.

Недоліки:

FactCheck.org спеціалізується переважно на факт-чекінгу політичних заяв, тому він може не покривати широкий спектр інших тем та галузей.

Проведення детального факт-чекінгу може займати час, тому вони можуть не завжди встигати реагувати на актуальні події в режимі реального часу.

1.3.2 Сервіс Snopes

Snopes – це інтернет-ресурс, спеціалізований на факт-чекінгу та перевірці правильності різних стверджень, чуток, міфів, фейк-новин та інших інформаційних тверджень. Заснований в 1995 році Девідом і Барбарою Міккелсен, Snopes виріс і став одним із найвідоміших та авторитетних джерел факт-чекінгу в інтернеті [21].

Основні аспекти діяльності Snopes включають наступне:

Факт-чекінг інформаційних тверджень: Snopes аналізує різні інформаційні твердження, починаючи від життєвих хаків і закінчуючи скандальними новинами та політичними заявами. Вони шукають докази та джерела, які підтверджують або спростовують те, що визначають як “правда” чи “брехня”.

Виявлення міфів та фейк-новин: Snopes спеціалізується на розкритті міфів, фейк-новин і чуток, що можуть бути широко поширеними в соціальних мережах та онлайн. Вони надають об'єктивну оцінку таких історій.

Аналіз ланцюгових листів і гасел: Snopes розслідує ланцюгові листи, гасла та інші локальні міфи, які можуть поширюватися через електронну пошту або соціальні мережі.

Переваги:

Snopes охоплює широкий спектр тем та інформаційних тверджень, від глобальних новин до загальноприйнятих переказів та популярних чуток. Ресурс славиться своєю об'єктивністю та незалежністю в оцінці інформації.

Snopes має велику базу даних з архівами факт-чекінгу, що дозволяє користувачам швидко знайти інформацію про різні теми.

Недоліки:

Також, як і багато інших ресурсів, Snopes може не завжди встигати реагувати на актуальні події в режимі реального часу.

Незважаючи на широкий охоплення, Snopes може не мати інформації щодо деяких менш відомих тем чи нових розслідувань.

1.3.3 Розширення NewsGuard

NewsGuard – це розширення для браузера та сервіс оцінки надійності новинарських джерел і видань, яке використовує штучний інтелект для оцінки та рейтингування джерел інформації [22]. Основною метою NewsGuard є надання користувачам інструментів для визначення достовірності новинних джерел та інформації, яку вони споживають в Інтернеті. Основні характеристики NewsGuard включають: NewsGuard надає кожному новинарському джерелу рейтинг на основі ряду критеріїв, включаючи точність, прозорість, відповідальність та інші. Це допомагає користувачам оцінювати джерела інформації. NewsGuard пропонує розширення для веб-браузерів, яке надає користувачам інформацію про достовірність сайтів, які вони відвідують. Користувачі можуть побачити рейтинг інтернет-джерела перед переходом на сайт. Якщо NewsGuard визначає, що новинарське джерело може містити дезінформацію або бути ненадійним, воно надає попередження користувачу і надає інформацію про причини такого рейтингу.

Висновки до розділу 1

У першому розділі було поставлено задачу дослідження виявлення дезінформації, де було вказано актуальність проблеми визначення не правдивої інформації під час воєнного стану. Також було розглянуто та обрано методи векторизації та класифікації тексту. Для векторизації обрано найсучасніший метод BERT, а саме SBERT, який дозволяє аналізувати контекст. Для класифікації буде використовуватись нейронна мережа XLNet. Датасетом будуть слугувати набори статтей з відомих журналів “The Guardian” та “New York Times” та дописи з соціальних мереж “Twitter” та “Reddit”. Було зроблено огляд існуючих рішень та дана характеристика переваг та недоліків.

2 МЕТОДИ ВЕКТОРИЗАЦІЇ ТА РОЗПІЗНАВАННЯ ДЕЗІНФОРМАЦІЇ В ТЕКСТОВИХ ПОВІДОМЛЕННЯХ

У другому розділі розглядаються методи векторизації та розпізнавання дезінформації в текстових повідомленнях. Аналізуються техніки передобробки тексту за допомогою RAKE, методи векторизації тексту в моделі SBERT та XLNet Large, а також методи класифікації в межах цих моделей. Висвітлюється процес прийняття рішення класифікації тексту на предмет наявності дезінформації, а також подається інформація щодо тренування моделей SBERT та XLNet Large.

2.1 Передобробка тексту за допомогою RAKE

RAKE (Rapid Automatic Keyword Extraction – Швидкий автоматичний пошук ключових слів) – це алгоритм для виділення ключових слів (слів-ключів) з текстових даних [23]. Основна ідея полягає в тому, щоб ідентифікувати слова, які часто зустрічаються разом, але можуть бути менш суттєвими, якщо розглядати їх окремо.

Використовуючи цей підхід, RAKE може виділити терміни, які представляють ключовий зміст тексту. Основна формула RAKE базується на визначенні ступеня слова у тексті, який визначається як сума частоти слова та відношення його ступеня до кількості слів у тексті. Важливими є також деякі параметри, такі як роздільники (знаки пунктуації, пробіли і т.д.) та ступінь слова у графі сусідства. Формула ступеня слова у тексті:

$$score(w) = F(w) + \frac{1}{\deg(w)} \sum_{v \in N(w)} F(v) \quad (2.1)$$

де, *w* – слово, *F(w)* – частота слова у тексті, *deg(w)* – ступінь слова у графі сусідства, *score(w)* – ступінь слова у тексті.

Ступінь слова у графі сусідства може бути визначений як кількість сусідів слова *w*. Таким чином, слова з високим *score(w)* вважаються ключовими словами за алгоритмом RAKE.

2.2 Векторизація тексту у моделі SBERT

Векторизація – це процес перетворення текстової інформації в числові вектори, які можуть бути використані для подальшого аналізу та взаємодії з моделями машинного навчання.

Щоб обробити текст природною мовою і витягти корисну інформацію з заданого слова або фрази за допомогою методів машинного навчання або глибокого навчання, необхідно перетворити рядок/текст у набір дійсних чисел (векторів) – вбудовування слів.

Вбудовування слів або векторизація слів – це методологія NLP, яка використовується для відображення слів або фраз у словнику у відповідні вектори дійсних чисел, а передбачення слів використовується для пошуку схожості/семантики слів.

Sentence-BERT – це підхід до отримання векторних представлень речень, який базується на трансформерній архітектурі та концепції подібності (similarity) [24]. Математична ідея полягає в тому, щоб забезпечити, щоб векторні представлення речень, отримані за допомогою кодування трансформера, були близькими в просторі векторів для семантично схожих речень [24].

Для кожного речення модель використовує трансформатор для отримання контекстуалізованого векторного представлення. У випадку SBERT, це векторне представлення містить інформацію про семантичний контекст слова у реченні.

Модель використовує механізм самоуваги для врахування контексту всередині речення та інших речень з набору даних.

Для векторизації слів у реченні трансформатор роздивляється кожне слово окремо, враховуючи його власний контекст у межах речення. Кожне слово отримує векторне представлення, яке залежить від його власного значення та значення інших слів у реченні.

Одним із ключових аспектів SBERT є контекстуалізація. Контекстуальні вектори отримуються з урахуванням контексту слова або речення в тексті. Це дозволяє моделі ефективно виражати семантичні зв'язки та розуміти значення слів у конкретному контексті.

Трансформер – це архітектура глибоких нейронних мереж, яка використовує механізм самоуваги для обробки та розуміння послідовностей даних, таких як тексти. Ця архітектура була вперше представлена у роботі “Attention is All You Need” [25]. Основною ідеєю трансформатора є використання механізму самоуваги для ефективного моделювання взаємозв'язків між елементами послідовностей.

У випадку Sentence-BERT, трансформатор використовується для отримання контекстуалізованих векторних представлень слів та речень. Контекстуалізовані вектори враховують синтаксичні та семантичні зв'язки між словами, що дозволяє моделі краще усвідомлювати значення та взаємодії між ними в контексті.

Механізм самоуваги дозволяє моделі визначати вагу різним частинам входу, залежно від їх контексту у реченні. Кожному слову у реченні призначається ваговий коефіцієнт (attention weight) α_{ij}, який вказує, наскільки важливим є дане слово для кожного іншого слова у реченні. Це дозволяє моделі фокусуватися на ключових елементах та їх взаємозв'язках, підвищуючи якість розуміння текстової інформації. Це обчислюється наступним чином:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{ik})} \tag{2.2}$$

де e_{ij} – значення, яке визначає, наскільки важливим є слово *j* для слова *i*.

Значення обчислюється за допомогою функції подібності між векторами q_i (запитання), k_j (ключ) та функції подібності *f:*

$$e_{ij} = f(q_i, k_j) \tag{2.3}$$

Отримані вагові коефіцієнти, які є результатом механізму самоуваги у трансформаторі, не лише служать для визначення важливості окремих слів у контексті, але й використовуються для обчислення зваженої суми векторів v_j (значення) для всіх слів *j* у реченні:

$$Attention(Q, K, V) = \sum_{j=1}^{n} \alpha_{ij} v_j \tag{2.4}$$

де *Q*, *K*, *V* – матриці запитань, ключів та значень, відповідно. Це операція самоуваги для слова *i*.

Отримана зважена сума використовується для контекстуалізації вектора h_i (вектора слова) у реченні:

$$Contextualized(h_i) = Attention(Q, K, V) \tag{2.5}$$

У моделі SBERT, трансформатор використовується як ключовий елемент для отримання контекстуалізованих векторних представлень для кожного слова в реченні. Цей процес відбувається на кількох рівнях абстракції, завдяки використанню кількох шарів трансформатора.

Кожен шар трансформатора відповідає за різні аспекти обробки інформації. Вони дозволяють моделі виявляти та ураховувати різні рівні абстракції та семантичні залежності в тексті. На кожному шарі трансформатора векторні

представлення слів отримуються через механізм самоуваги, який надає моделі змогу зосереджувати увагу на різних аспектах контексту.

Ця багаторівнева архітектура трансформера дозволяє моделі узагальнювати контекстуальні взаємодії на різних рівнях, починаючи від простих синтаксичних зв'язків і закінчуючи більш складними семантичними структурами. Кінцевий вектор для кожного слова, отриманий з трансформатора, виступає як контекстуалізоване векторне представлення цього слова, яке враховує його значення в конкретному контексті речення. Такий підхід дозволяє моделі SBERT ефективно враховувати семантичні відносини та контекстуальні аспекти тексту, покращуючи якість обробки природної мови.

2.3 Методи класифікації у моделі SBERT

В констрастивному навчанні, яке використовує SBERT, ключовим аспектом є підхід, що базується на порівнянні подібності між векторними представленнями речень. Основна мета цього підходу – забезпечити те, щоб вектори, що відповідають семантично подібним реченням, були близькими один до одного у просторі векторів, тоді як вектори різних речень, навпаки, повинні відділятися.

Це досягається шляхом максимізації подібності (зазвичай використовується косинусна подібність) між векторами, що відповідають позитивним парам речень, тобто парам речень, які семантично схожі. У той же час, система намагається мінімізувати подібність між векторами негативних пар речень, тобто речень, які не пов'язані за змістом. Для двох векторів u та v подібність може бути визначена за допомогою косинусної подібності:

$$similarity(u, v) = \frac{u \cdot v}{\|u\| \cdot \|v\|} \tag{2.6}$$

де $u \cdot v$ – скалярний добуток, а $\|u\|$ та $\|v\|$ – норми векторів.

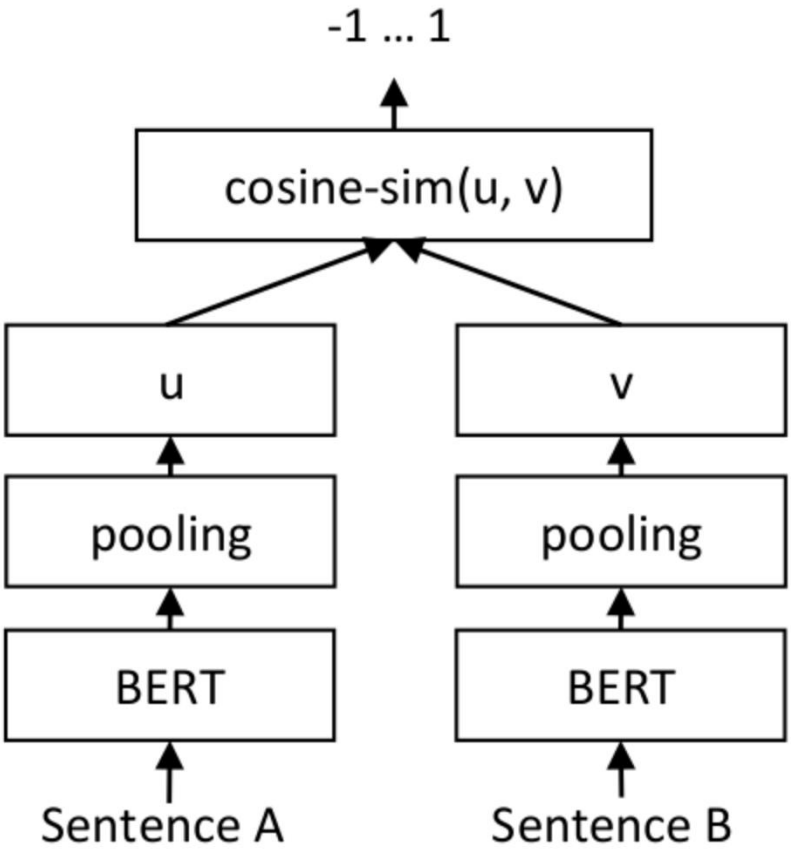

Рисунок 2.1 – Архітектура пошуку подібності між двома реченнями у моделі SBERT

Для пар речень (запитання, текст) обчислюється подібність між векторами речень і визначається функція втрати.

Нехай *Q* та *D* – вектори значень (Query та Document), а *sim(Q, D)* – їх косинусна подібність.

Функція втрати SBERT має дві складові – для позитивних та негативних пар. Позитивні пари відображають схожість між реченнями, а негативні пари генеруються випадковим вибором інших, не пов'язаних за змістом, речень.

Мета функції втрати полягає в тому, щоб мінімізувати відстань між векторами позитивних пар, тобто речень, які мають схожий семантичний зміст, та одночасно максимізувати відстань між векторами негативних пар, представляючи неподібні речення. Це забезпечує, що модель ефективно вивчає семантичні відносини між реченнями та створює векторні представлення, які точно відображають схожість чи відмінність між ними. Такий підхід глибше вбирає семантичний контекст та покращує якість векторних представлень в роботі з текстовими данними. Формула для функції втрати в SBERT може виглядати наступним чином:

$$L(Q, D, Y) = -\frac{1}{N}\sum_{i=1}^{N}\left[Y_i \cdot \log\left(sim(Q_i, D_i)\right) + (1 - Y_i) \cdot \log\left(1 - sim(Q_i, D_i)\right)\right] \qquad (2.7)$$

де N – кількість пар речень, Y_i – мітка класу (1 для позитивної пари, 0 для негативної пари), $sim(Q_i, D_i)$ – косинусна подібність між векторами Query та Document для *i*-тої пари.

Процес контрастивного навчання в SBERT складається з формування позитивних та негативних пар. Для кожного речення вибирається позитивне речення (запитання або текст), яке є семантично подібним. Та для кожного речення вибирається негативне речення, яке є семантично віддаленим (зазвичай випадковим чином обирається інше речення). Наступним кроком є обчислення витрат, а саме обчислюється косинусна подібність між парами речень, після обчислюється функція витрати для кожної пари. Останнім кроком є застосування градієнтного спуску для покращення векторних представлень і зменшуючи значення функції витрати. Таким чином, контрастивне навчання в SBERT забезпечує отримання векторних представлень речень, які ефективно взаємодіють із семантичною схожістю та різницею між реченнями.

Ці векторизовані представлення можуть бути використані для різних задач, включаючи вимірювання схожості, класифікацію тексту, або пошук відповідей.

Важливою особливістю SBERT є те, що вектори для запитань та відповідей можуть бути схожими, що дозволяє використовувати їх для пошуку відповідей у тексті.

Векторизація у SBERT дозволяє враховувати глобальний контекст у тексті, що поліпшує семантичну розуміність. Значення векторів не обмежуються тільки локальним контекстом слова чи речення, але враховують усі дані з корпусу текстів, що дозволяє моделі ефективно вирізняти семантичні особливості слів та речень у різних контекстах.

Знаючи цю інформацію, ми можемо обирати топ-*K* семантично схожих речень для подальшого порівняння та знаходження правдивих та неправдивих тверджень. В цьому допоможе друга нейронна мережа – XLNet Large.

2.4 Векторизація тексту у моделі XLNet Large

Токенізація представляє собою важливий етап обробки тексту, який полягає в розбитті вхідного тексту на індивідуальні токени або підрядки. У випадку моделі XLNet Large, застосовується розширена та контекстуалізована токенізація, спрямована на ефективну обробку текстової інформації з урахуванням його структури та семантики [26].

Токен в цьому контексті є найменшою одиницею тексту, яку модель може обробити. Токен може включати в себе слова, підслова або символи, в залежності від використовуваної токенізації. У моделі XLNet Large використовуються базові токени, які можуть представляти слова, числа або пунктуаційні елементи, і кожному токену надається унікальний ідентифікатор.

Розширена токенізація включає в себе детальний розбір тексту на токени, що можуть включати в себе слова, пунктуацію та інші структурні компоненти. Цей підхід дозволяє зберегти важливу інформацію про текст і його структуру, що

важливо для подальшого ефективного використання моделі в різних завданнях обробки природної мови. Формула для розрахунку ідентифікатора токена *token_id*:

$$token_id = Tokenizer(t) \quad (2.8)$$

де *t* – токен, *Tokenizer(t)* – функція токенізатора, яка призначає унікальний ідентифікатор кожному токену *t*.

Контекстуалізована токенізація враховує контекст кожного токена в тексті. Оскільки значення токена може залежати від його контексту, контекстуалізована токенізація надає додаткову інформацію для створення контекстуалізованих представлень.

Кожен токен обробляється з урахуванням його контексту в тексті, що враховує семантичний та граматичний контекст. Контекстуалізоване представлення токена *token_embedding* обчислюється за допомогою контексту та попередніх токенів у тексті:

$$token_embedding = EmbeddingFunction(t, context_i) \quad (2.9)$$

де, *t* – токен, $context_i$ – контекст токена, який може включати попередні токени у тексті перед *t*.

Типова EmbeddingFunction (EFunction) може бути описана наступним чином:

$$EFunction = TransformerEncoder\big(Embed(t) + ContextEmbed(context_i)\big) \quad (2.10)$$

де t – токен, $context_i$ – контекст токена, який може включати попередні токени у тексті перед *t*, *Embed(t)* – одновимірний вектор, представляючи токен *t*, *ContextEmbed(*$context_i$*)* – одновимірний вектор, представляючий контекст токена.

TransformerEncoder – трансформаторний енкодер, який обробляє об'єднання вбудовувань токена і контексту для отримання контекстуалізованого векторного представлення.

Для кожної послідовності токенів додаються спеціальні маркери [CLS] та [SEP], які вказують на початок і кінець послідовності.

$$TokenizedText = [CLS] + tokens + [SEP] \quad (2.11)$$

Такий підхід дозволяє моделі правильно визначити границі тексту та врахувати його структуру. У введенні використовується маска уваги для вказівки моделі, які токени враховувати та які ігнорувати. Маска уваги генерується на основі послідовності токенів та маркерів [PAD], які додаються для досягнення однакової довжини введення:

$$AttentionMask[i] = \begin{cases} 1 \; if \; tokenizedText[i] \neq PAD \\ 0 \; if \; tokenizedText[i] = PAD \end{cases} \quad (2.12)$$

Ці параметри та функції є критичними для векторизації текстових даних та створення подальших контекстуалізованих представлень для використання моделлю XLNet Large. Вони дозволяють ефективно обробляти тексти та забезпечують моделі інформацію про контекст кожного токена.

2.5 Методи класифікації у моделі XLNet Large

Модель XLNet Large використовує авторегресивний підхід для генерації тексту, що означає, що кожен токен у текстовій послідовності генерується з урахуванням контексту та попередніх токенів [26]. Цей підхід дозволяє

створювати текст з логічною структурою та враховувати зв'язок між різними частинами тексту.

Формула ймовірності тексту визначає, наскільки ймовірно весь текст у вибірці. Ми обчислюємо ймовірність кожного токена, враховуючи контекст всіх попередніх токенів у тексті. Множення цих ймовірностей дає загальну ймовірність тексту.

$$P(sequence) = \prod_{t=1}^{T} P(token_t | context_{<t}) \quad (2.13)$$

де *P(sequence)* – ймовірність всієї послідовності тексту, T – кількість токенів у тексті, $P(token_t | context_{<t})$ – умовна ймовірність генерації конкретного токена у позиції t при врахуванні контексту та попередніх токенів.

Необхідно максимізувати ймовірність тексту, шукаючи оптимальні параметри θ, які максимізують значення ймовірності $P(sequence|\theta)$.

$$\theta^* = argmax_\theta P(sequence|\theta) \quad (2.14)$$

де θ – параметри моделі, θ^*- оптимальні параметри, які максимізують ймовірність генерації тексту.

Модель XLNet Large використовує перестановки для моделювання взаємозв'язків між токенами в тексті. Цей механізм дозволяє моделі враховувати не лише порядок слів, а й їх взаємодію та залежність одне від одного.

В контексті моделі XLNet Large, перестановка визначається як випадковий порядок токенів у тексті. Кожен токен отримує свій векторне представлення, і їх порядок в тексті обирається випадковим чином.

Модель обчислює ймовірність кожної можливої перестановки токенів (14) у тексті. Використовуючи ці ймовірності, вона враховує всі можливі порядки слів під час генерації тексту:

$$P(permutation) = \frac{1}{N!} \quad (2.15)$$

де *P(permutation)* – ймовірність конкретної перестановки, *N!* – факторіал кількості токенів у тексті.

Цей механізм дозволяє XLNet Large моделювати більш складні взаємозв'язки між словами в тексті та поліпшує здатність моделі до генерації тексту з урахуванням різних порядків слів.

Виходом моделі є вектор значень ймовірностей приналежності до одного з трьох класів $[P_{entailment}, P_{neutral}, P_{contradiction}]$.

2.6 Прийняття рішення класифікації тексту на наявність дезінформації

В процесі обробки речень за допомогою SBERT та XLNet Large, наша модель генерує вектори ймовірностей для кожного з трьох класів: $[P_{entailment}, P_{neutral}, P_{contradiction}]$. Ці вектори відображають ступінь належності кожного речення до відповідного класу, і вони представляють значущість кожної категорії у контексті задачі класифікації.

Розглядаючи SBERT, ця модель базується на архітектурі BERT (Bidirectional Encoder Representations from Transformers), яка дозволяє враховувати контекст тексту в обидві сторони слова при генерації векторів представлення. Це дозволяє отримати більш точні та контекстуалізовані вектори для подальшої класифікації. З іншого боку, XLNet Large використовує трансформерну архітектуру, забезпечуючи подовжену залежність між словами та більший контекст для кращого розуміння тексту.

Коли отримано вектори ймовірностей від обох моделей для кожного речення, проводиться ансамблювання. Цей процес включає у себе обчислення

середнього значення відповідних ймовірностей для кожного класу. Таким чином, отримується новий вектор ймовірностей для кожного класу, який представляє узагальнену інформацію від обох моделей.

Остаточний етап включає використання методу більшості голосів (majory voting) для визначення кінцевого класу для кожного речення. Знаходження максимальної ймовірності серед об'єднаних векторів дозволяє нам визначити, максимальне значення кожного з класів.

$$predLabel = argmax_i \sum_{j=1}^{N} P_{ij} \quad (2.16)$$

де *predLabel* – кінцевий клас, який ми визначаємо за допомогою голосування більшості, *i* – індекс класу (Entailment, Neutral, Contradiction), *j* – індекс речення в наборі даних, *N* – загальна кількість речень в наборі даних, P_{ij} – ймовірність належності речення *j* до класу *i*, яку ми отримали під час ансамблювання векторів ймовірностей.

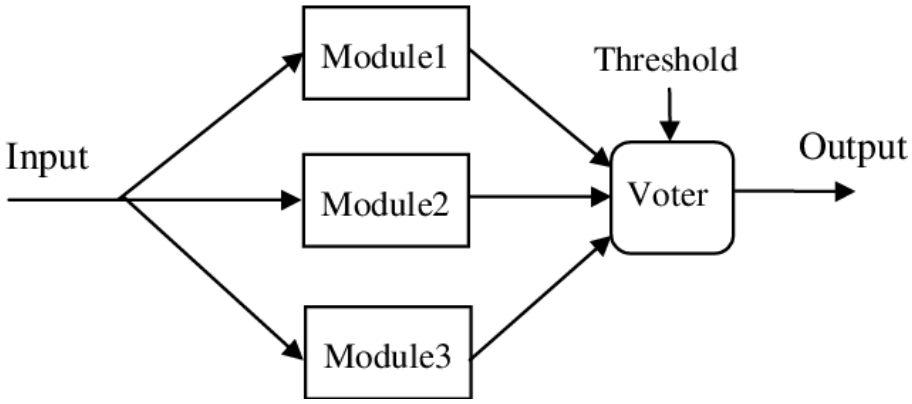

Рисунок 2.2 – Блок-схема методу більшості голосів

Результатом є:

$$normscore = [max(P_{entailment}), max(P_{neutral}), max(P_{contradiction})]. \quad (2.17)$$

Далі розраховується значення дизінформації у відсотках (18):

$$disinfoScore = \begin{cases} 50 + \frac{normscore[2] - normscore[0]}{2} \cdot 100 \;\; if\ predLabel = 0 \\ 50 + \frac{normscore[0] - normscore[2]}{2} \cdot 100 \;\; if\ predLabel = 2 \\ 50 + \frac{normscore[2] - normscore[0]}{2} \cdot 100 \;\; if\ predLabel = 0 \end{cases} \quad (2.18)$$

де *predLabel* – передбачений клас моделі для вхідного тексту, *normscore[0]*, *normscore[2]* – нормалізовані значення ймовірностей для класу 0 (entailment) та класу 2 (contradiction), які було обчислені попередньо, *disinfoScore* – результуючий показник деінформаційності, де 0 вказує на entailment та 100 на contradiction.

Цей підхід дозволяє ефективно використовувати інформацію від обох моделей для поліпшення точності та надійності класифікації текстових даних. Використовуючи ансамбль моделей та метод більшості голосів, ми створюємо надійний механізм класифікації, який враховує сильні сторони кожної з моделей.

2.7 Тренування моделі SBERT

На першому кроці алгоритм випадково ініціалізує параметри моделі. Це означає, що початкові значення параметрів моделі вибрані випадковим чином.

Параметри моделі – це значення, які контролюють поведінку моделі. Вони використовуються для опису того, як модель обробляє інформацію і як вона приймає рішення.

Існує кілька різних способів випадкової ініціалізації параметрів моделі. Один із поширених способів – використання генератора випадкових чисел.

Генератор випадкових чисел генерує числа, які розподілені рівномірно в певному діапазоні.

Інший поширений спосіб випадкової ініціалізації параметрів моделі – використання нормального розподілу. Нормальний розподіл – це розподіл ймовірностей, який часто зустрічається в природі.

Який спосіб випадкової ініціалізації параметрів моделі використовувати, залежить від конкретної моделі та задачі, яку вона вирішує.

Випадкова ініціалізація параметрів моделі допомагає уникнути локальних мінімумів. Локальні мінімуми – це точки, в яких функція втрат не може бути зменшена далі без зміни напрямку оптимізації.

Випадкова ініціалізація параметрів моделі допомагає моделі генерувати більш різноманітні результати.

На другому кроці алгоритм пропускає речення x_i через модель SBERT. Це означає, що модель отримує речення x_i як вхід і повертає значення $f(x_i)$ як вихід. Значення $f(x_i)$ є оцінкою подібності між реченням x_i та x_i'.

На третьому кроці алгоритм обчислює значення функції втрат L. Функція втрат – це число, яке оцінює, наскільки добре модель SBERT може передбачити подібність між двома реченнями. У випадку моделі SBERT функція втрат є ймовірнісною функцією втрат. Це означає, що вона обчислює ймовірність того, що модель SBERT правильно передбачить подібність між двома реченнями.

На четвертому кроці алгоритм обчислює градієнт функції втрат щодо параметрів моделі. Градієнт – це вектор, який показує, в якому напрямку потрібно змінити параметри моделі, щоб зменшити значення функції втрат.

У випадку моделі SBERT градієнт функції втрат обчислюється за допомогою методу назадного поширення. Метод назадного поширення – це алгоритм, який використовується для обчислення градієнтів функцій втрат для нейронних мереж.

Алгоритм назадного поширення починається з кінця моделі SBERT. Це означає, що алгоритм починає з вихідного вектора моделі. Вихідний вектор моделі – це вектор, який виходить з моделі SBERT. Він використовується для

оцінки подібності між двома реченнями. На другому кроці алгоритм обчислює градієнт функції втрат щодо вихідного вектора моделі. Градієнт – це вектор, який показує, в якому напрямку потрібно змінити вихідний вектор моделі, щоб зменшити значення функції втрат.

Третій крок алгоритму назадного поширення полягає в обчисленні градієнта вихідного вектора моделі щодо внутрішнього параметра моделі. Внутрішні параметри моделі – це значення, які контролюють поведінку моделі. Градієнт вихідного вектора моделі щодо внутрішнього параметра моделі показує, як зміна внутрішнього параметра моделі вплине на вихідний вектор моделі.

Четвертий крок алгоритму назадного поширення полягає в тому, що алгоритм повертається вгору по моделі SBERT, обчислюючи градієнти функції втрат щодо внутрішніх параметрів моделі. Цей процес повторюється, поки алгоритм не дійде до початку моделі SBERT.

П'ятий крок алгоритму назадного поширення полягає в тому, що алгоритм оновлює параметри моделі, використовуючи градієнти, обчислені на попередньому кроці. Оновлення параметрів моделі здійснюється за допомогою наступної формули:

$$\omega \leftarrow \omega - \alpha \nabla_{\omega} L \tag{2.19}$$

де ω – внутрішні параметри моделі, α – крок оновлення, $\nabla_{\omega} L$ – градієнт функції втрат щодо внутрішнього параметра моделі.

2.8 Тренування моделі XLNet Large

Перший крок – завантаження набору даних для тренування. XLNet Large може бути навчений на будь-якому наборі даних, який містить текстові пари. Для

прикладу, можна використовувати набори даних для задач NLP, таких як переклад, класифікація або відповіді на запитання.

Наступний крок – ініціалізація параметрів моделі. XLNet Large використовує випадкову ініціалізацію параметрів.

Попереднє тренування – це процес, який допомагає моделі навчитися базовим задачам, таким як розпізнавання слів та граматики. XLNet Large використовує попереднє тренування за допомогою задачі попереднього тренування трансформера. Задача попереднього тренування трансформера полягає в тому, щоб навчити модель генерувати наступний токен в рядку тексту. Для цього модель отримує на вхід послідовність токенів і повинна генерувати наступний токен.

XLNet Large виконує попереднє тренування протягом 100 епох.

Основне тренування – це процес, який допомагає моделі навчитися виконувати конкретні завдання, такі як переклад, класифікація або відповіді на запитання. XLNet Large використовує основне тренування за допомогою задачі XLNet.

Висновки до розділу 2

У другому розділі було розглянуто математичне підґрунтя та методи реалізації векторизації та розпізнавання тексту дезінформації. Було визначено переваги сучасних нейромереж, а також запропоновано метод ансамблювання двох моделей: Sentence-BERT для виявлення топ-K кандидатів з бази даних за схожістю та XLNet Large для виявлення належності до одного з трьох класів: entailment, neutral та contradiction.

У результаті дані показники використовується для визначення фінального вердикту дезінформації тексту, де значення варіюються від 0 до 100.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ МЕТОДУ ВИЯВЛЕННЯ ДЕЗІНФОРМАЦІЇ В ТЕКСТАХ ЗМІ

У третьому розділі презентується програмна реалізація методу виявлення дезінформації в текстах ЗМІ. Починаючи з вибору технологій для бекенду, розглядаються аспекти використання мови програмування Python, фреймворку FastAPI для інтеграції з фронтенд-частиною, та бібліотек для обробки тексту природньої мови, зокрема SBERT та XLNet Large для семантичної обробки текстів. Далі висвітлюється вибір технологій для реалізації фронтенду, включаючи мову програмування JavaScript, фреймворк React, бібліотеки для дизайну та стилізації, а також для візуалізації даних. Висвітлюється реалізація інтерфейсу користувача при вході, відображення графіків датасету, введення даних та їхнє відображення, а також аналітика по виявленню дезінформації, яка включає в себе відображення ключових слів та підсумок роботи аналітика.

3.1 Вибір технологій для реалізації бекенду

Наукове дослідження в області вибору мов програмування є важливою сферою, оскільки це визначає не лише технічні аспекти розробки, але і впливає на продуктивність та перспективи подальшого розвитку програмних проєктів. Врахування різноманітних аспектів допомагає здійснити інформований вибір для досягнення оптимальних результатів у розробці програмного забезпечення.

Оскільки проєкт має бути сучасним, необхідно розглянути всі високорівневі інструменти програмування та проектування.

3.1.1 Мова програмування Python

Швидкодія мови програмування визначається кількома ключовими факторами, такими як метод компіляції чи інтерпретації та споживання ресурсів. Мови, що компілюються, зазвичай забезпечують ефективнішу швидкодію, але важливо враховувати ймовірні витрати під час розробки. Отже, баланс між швидкодією та зручністю для програміста є ключовим аспектом при виборі мови.

Легкість у використанні мови програмування визначається її синтаксисом та наявністю розширених інструментів. Мови з простим синтаксисом, такі як Python, забезпечують швидшу розробку та підтримують легке розуміння програмного коду. Доступність багатьох бібліотек та фреймворків також впливає на легкість розробки та розширення функціональності програми.

Розуміння науковою спільнотою є критичним елементом вибору мови програмування через вплив на підтримку, обмін досвідом та використання в наукових дослідженнях. Активність та експертність спільноти розробників дозволяє ефективніше вирішувати проблеми та вдосконалювати розроблені рішення.

Екосистема та підтримка мови програмування визначаються наявністю інструментів розробки, які полегшують та прискорюють процес програмування. Постійна підтримка та регулярні оновлення сприяють стабільності та актуальності вибраної мови в середовищі розробки.

Врахування специфікацій конкретного проєкту, вимог до продукту та експертиза команди розробників є необхідними факторами визначення оптимальної мови програмування для конкретного випадку. Врахування цих аспектів дозволяє забезпечити ефективну та успішну розробку програмного забезпечення відповідно до потреб проєкту.

Враховуючи різноманіття критеріїв, які визначають оптимальний вибір мови програмування, важливо звернути увагу на індекс TIOBE [27]. Цей індекс відображає популярність мов програмування на основі різноманітних факторів, включаючи кількість розробників, публікацій та проєктів у сфері програмування.

Індекс використовується для прийняття рішення про те, яку мову програмування слід обрати при створенні нової програмної системи.

Таблиця 3.1 – Рейтинг мов програмування за індексом TIOBE

Nov 2023	**Nov 2022**	**Programming Language**	**Ratings**	**Change**
1	1	Python	14.16%	-3.02%
2	2	C	11.77%	-3.31%
3	4	C++	10.36%	-0.39%
4	3	Java	8.35%	-3.63%
5	5	C#	7.65%	3.40%

Найкращим вибором є мова програмування Python [28]. Ця мова відзначається простим синтаксисом, що полегшує розробку та розуміння коду, що є важливим фактором для ефективної роботи над проєктом. Легкість у використанні Python підтримується широким спектром бібліотек і фреймворків, що сприяє швидкому розгортанню та розширенню функціональності програми.

Python також володіє активною та експертною спільнотою розробників, що робить його відмінним вибором для нашого проєкту. Ця спільнота забезпечить підтримку, обмін досвідом та відповіді на можливі труднощі під час розробки.

Однак найбільш важливим фактором у виборі Python є його потужність та ефективність в області обробки тексту, донавчання та класифікації.

Python має ряд високопродуктивних бібліотек, таких як TensorFlow та PyTorch, які дозволяють легко реалізувати та навчати моделі машинного навчання для класифікації текстових даних.

Це робить Python ідеальним вибором для нашого завдання, пов'язаного з аналізом тексту на наявність дизінформації.

3.1.2 Фреймворк інтеграції з фронтенд-частиною FastAPI

FastAPI [29] – це сучасний веб-фреймворк для створення веб-додатків на мові програмування Python, який спеціалізується на швидкості і типізації. Основні переваги включають високу швидкість виконання за рахунок використання стандарту ASGI, автоматичну документацію Swagger UI та ReDoc на основі анотацій типів даних, статичну типізацію Python для полегшення автоматичної перевірки введення та виведення даних, асинхронне програмування для ефективної обробки багатьох запитів одночасно, вбудовану підтримку стандартів безпеки, таких як OAuth2 і JWT, підтримку відкритих стандартів для інтеграції з іншими службами та простий та зрозумілий інтерфейс для швидкої розробки веб-додатків, зберігаючи високий рівень продуктивності.

3.1.3 Бібліотеки для попередньої обробки тексту природньої мови

NLTK (Natural Language Toolkit) [30] представляє собою впливову бібліотеку, призначену для розширених завдань обробки тексту. Забезпечуючи інструменти для токенізації, лематизації та аналізу синтаксису, вона дозволяє вченим та розробникам реалізовувати складні алгоритми обробки природної мови.

Gensim, у свою чергу, вирізняється своєю здатністю до тематичного моделювання та векторної індексації текстових даних. Вона дозволяє виконувати аналіз зв'язків між словами та концептуалізовувати великі об'єми текстової інформації.

Emoji Library стає невід'ємним елементом у сфері обробки текстових даних, забезпечуючи інструменти для роботи з емодзі. Це особливо важливо в контексті аналізу текстів з емоційним забарвленням або спілкування в соціальних мережах.

Обрані бібліотеки разом створюють потужний інструментарій для вчених та розробників, який полегшує завдання в області обробки текстової інформації, сприяючи вивченню та розумінню великих обсягів текстових даних.

3.1.4 Бібліотека для семантичної обробки текстів природньої мови SBERT

У світі аналізу текстових даних та обробки природної мови, бібліотека Sentence Transformers (SBERT) [31] визначається не лише своєю передовою технологічною основою, але й потужністю її застосування в екосистемі мови програмування Python.

SBERT, заснована на трансформерній архітектурі та використовуючи модель BERT, стала синонімом векторизації текстів з врахуванням семантичних зв'язків. Вона не просто генерує вектори для слів та фраз, але створює простір векторів, в якому семантично близькі текстові фрагменти отримують схожі векторні представлення.

Важливим аспектом її застосування в Python є не лише легка інтеграція завдяки багатофункціональному пакету мови, але й потужність цієї бібліотеки в поєднанні з іншими інструментами для обробки тексту в цьому середовищі. Це відкриває широкі можливості для вивчення семантики, кластеризації та класифікації текстових даних, забезпечуючи високий рівень точності та ефективності в різноманітних завданнях.

3.1.5 Бібліотека для семантичної обробки текстів природньої мови XLNet Large

Додавання великої моделі XLNet Large до інструментарію відкриває нові перспективи в розумінні та обробці текстових даних: Ця вдосконалена архітектура, розроблена спільнотою Hugging Face [32], надає важливі додаткові функції для задач розуміння природної мови.

XLNet Large не обмежується однобічним навчанням, а вміє аналізувати тексти в багатонапрямленому форматі. Використовуючи контекстні ембедінги, ця

модель досягає глибшого розуміння зв'язків у текстах, що робить її незамінною для вирішення великої кількості завдань в галузі обробки природної мови.

3.2 Вибір технологій для реалізації фронтенду

Вибір технологій для розробки фронтенду був здійснений на основі ретельного аналізу вимог та покладених завдань, а також з урахуванням сучасних тенденцій у сфері веб-розробки. Цей вибір є результатом науково обгрунтованого підходу і має ряд об'єктивних причин. Обрані технології взаємодіють між собою, утворюючи збалансоване технічне стекове рішення, яке відповідає сучасним стандартам веб-розробки та гарантує якість та ефективність реалізації.

3.2.1 Мова програмування JavaScript

JavaScript [33] є важливою мовою програмування для розробки інтерфейсів завдяки своїй універсальності та здатності створювати динамічні, інтерактивні веб-сторінки. JavaScript використовується для програмування веб-додатків на стороні клієнта, дозволяючи користувачам взаємодіяти з вмістом сторінки без необхідності перезавантаження сторінки. JavaScript використовується для клієнтського програмування веб-додатків, що дозволяє користувачам взаємодіяти з вмістом сторінки без необхідності перезавантаження сторінки.

JavaScript – це інтерпретована мова, яка виконується безпосередньо в браузері, що дозволяє йому реагувати на події, обробляти дані і змінювати структуру сторінки в режимі реального часу. Це робить її ідеальним інструментом для створення живих і динамічних користувацьких інтерфейсів.

Використання модульності в JavaScript особливо важливе для підтримки великих і складних веб-додатків. Модульність дозволяє розбивати код на менші незалежні частини, якими можна легко керувати і відокремлювати одна від одної, забезпечуючи чистий і організований код.

JavaScript і Python можуть взаємодіяти в різних сценаріях, що робить їх потужною комбінацією для розробки повноцінних веб-додатків.

3.2.2 Фреймворк створення інтерфейсів React

React [34] – це бібліотека для розробки користувацьких інтерфейсів, розроблена компанією Facebook, яка фокусується на концепції компонентно-орієнтованого програмування. Цей підхід дозволяє розбивати інтерфейси на невеликі самодостатні компоненти, які можна незалежно розробляти, тестувати та підтримувати.

Основна ідея компонентного програмування в React полягає у створенні багаторазових і керованих частин інтерфейсу. Компоненти можуть бути простими елементами (наприклад, кнопками) або складними блоками (наприклад, формами і таблицями). Кожен компонент може мати власні стани та реквізити, що полегшує передачу та керування даними між компонентами.

Щоб швидко реагувати на зміни в даних, React використовує віртуальну DOM (Document Object Model). Віртуальний DOM є абстракцією реального DOM і дозволяє React оптимізувати процес оновлення сторінки, обчислюючи лише зміни, а не оновлюючи всю сторінку. Це робить React дуже ефективним при оновленні користувацького інтерфейсу, особливо коли є великі обсяги даних або інтенсивна взаємодія з користувачем.

React також підтримує одностороннє зв'язування даних, що полегшує прозорий і передбачуваний обмін даними між компонентами. Коли стан компонента змінюється, React автоматично повторно рендерить лише необхідні

частини інтерфейсу, що забезпечує оптимальне використання ресурсів та підвищує продуктивність додатку.

Ці особливості роблять React чудовим інструментом для розробки користувацьких інтерфейсів, забезпечуючи легкість, ефективність і простоту розробки та супроводу великомасштабних веб-додатків.

3.2.3 Бібліотеки реалізації дизайну та стилізації Bootstrap

У сучасній веб-розробці використання фреймворків для стилізації компонентів є важливим елементом, спрямованим на полегшення роботи розробника та забезпечення послідовного і красивого дизайну веб-додатків. Фреймворки Bootstrap [35] і Reactstrap [36] є популярними інструментами, призначеними для вирішення завдань стилізації та надання веб-сторінкам сучасного вигляду.

Bootstrap відрізняється своєю повнотою і різноманітністю готових компонентів, які дозволяють розробникам швидко і ефективно створювати різні елементи інтерфейсу. Його адаптивний дизайн сприяє оптимальному відображенню сторінок на різних пристроях, що є важливим фактором у світі мобільних технологій.

Reactstrap, з іншого боку, є обгорткою, розробленою спеціально для Bootstrap, що дозволяє легко інтегрувати його компоненти в середовище додатків React. Це дозволяє розробникам використовувати готові Bootstrap-дизайни так, ніби вони є нативними React-компонентами.

Використання цих фреймворків робить процес стилізації інтерфейсу та розробки більш ефективним і менш трудомістким. Розробники можуть зосередитися на логіці додатку, не випускаючи з уваги важливість користувацького досвіду.

Інтеграція цих фреймворків у розробку веб-додатків розширює технічні можливості і дозволяє створювати високоякісні веб-інтерфейси, які відповідають сучасним стандартам і вимогам користувачів. Такий підхід дозволяє ефективно розробляти, підтримувати та оптимізувати веб-додатки у швидкозростаючому інтернет-середовищі.

3.2.4 Бібліотеки для візуалізації даних

У контексті веб-розробки використання бібліотек для візуалізації даних є важливим елементом, спрямованим на покращення сприйняття інформації користувачами. Highcharts [37] – це бібліотека для створення діаграм, що характеризується високою гнучкістю та широким спектром можливостей візуалізації. Бібліотека надає розробникам інструменти для ефективного відображення числових даних у вигляді лінійних графіків, кругових діаграм, стовпчиків та інших візуальних елементів.

React-wordcloud [38] представляє новий аспект візуалізації через концепцію хмар слів. Цей підхід дозволяє створювати естетичні та інформативні відображення, виділяючи ключові терміни та поняття на основі їх частоти в текстових даних.

Бібліотека React-toastify може бути використана для відображення повідомлень користувачам в інтерактивному форматі. Це може допомогти тримати користувачів в курсі важливих подій, що відбуваються в додатку, і створити позитивний користувацький досвід.

Поєднання цих технологій не тільки дозволяє візуалізувати числові дані в графічній формі, але й додає рівень інтерактивності та наочності для користувача. Такий підхід не лише сприяє ефективній інформаційній комунікації, але й створює зацікавленість та залучення користувача до процесу взаємодії з веб-додатком.

3.3 Збір та індексація даних

Формування датасету для моделі розпізнавання фактів є важливим кроком у процесі її розробки. Датасет повинен бути репрезентативним для реального світу та містити достатню кількість інформації для навчання моделі. Також важливо проіндексувати дані, щоб мати до них швидкий доступ.

3.3.1 Пошуковий двигун ElasticSearch

Elasticsearch [39] – це потужна розподілена система пошуку та аналізу даних, що виникла як частина проекту Apache Lucene. Розроблений для швидкого і легкого пошуку великих обсягів даних, Elasticsearch надає природний інтерфейс для роботи зі структурованими і неструктурованими даними в режимі реального часу.

Завдяки своїй гнучкості та неперевершеній швидкості для всіх типів даних, Elasticsearch використовується в різних галузях, від пошуку та аналізу журналів до побудови систем обробки текстової інформації.

Важливою складовою ElasticSearch є можливість створювати індекси, які значно полегшують організацію та пошук даних. Розподілена архітектура пошуку забезпечує високу доступність і масштабованість системи та дозволяє обробляти великі обсяги інформації.

Однією з найважливіших переваг ElasticSearch є його здатність обробляти складні запити з точно налаштованими критеріями пошуку та аналізу. Це робить його ідеальним інструментом для вирішення широкого спектру завдань, таких як побудова інтелектуальних систем обробки тексту.

Вся екосистема Elastic Stack, включаючи Logstash і Kibana, робить технологію більш комплексною і простою в розгортанні та управлінні.

ElasticSearch продовжує залишатися в авангарді рішень для обробки тексту та аналітики завдяки своїм вражаючим можливостям і надійності.

В ElasticSearch індекс представляє собою структуру, яка дозволяє швидко та ефективно виконувати пошук та аналіз даних. Індекс відображається у вигляді директорії, яка містить у собі рядки, названі “документами”. Кожен документ в індексі включає дані та відомості, що дозволяють еластичному пошуковому двигуну знаходити та звертатися до цих даних ефективно.

Структура індексу може бути визначена мапінгом, який описує, які поля містить кожен документ та який тип даних у кожному полі. ElasticSearch може автоматично створювати індекси або вони можуть бути створені та налаштовані користувачем.

3.3.2 Збір еталонних текстів природньої мови у датасет

Для забезпечення ефективної роботи моделі розпізнавання фактів в контексті війни в Україні, формування датасету включало кілька ключових аспектів.

В першу чергу, датасет був націлений на включення статей, які тісно пов'язані з війною в Україні. Це стратегічне рішення спрямоване на навчання моделі розпізнавати факти, які важливі в контексті даної теми.

Для забезпечення репрезентативності датасету були використані матеріали з різних джерел. Зокрема, в датасет включено статті з The New York Times та The Guardian, які відомі своєю об'єктивністю та чіткістю висвітлення подій.

Особлива увага була приділена актуальності даних. Вибірка статей охоплює період з 24 лютого 2022 року по 15 березня 2022 року, що враховує весь період війни в Україні. Це забезпечує актуальність та відповідність отриманого датасету.

Для збору та обробки великого обсягу даних був використаний спеціальний парсер (див. Додаток А), який дозволив систематизувати необхідну інформацію для подальшого використання в розробці моделі розпізнавання фактів.

Таблиця 3.2 – Опис зібраних полів статтей

Параметр	**Опис**
resource_id	Назва веб-сайту, на якому була опублікована стаття
head	Заголовок статті
content	Текст статті

Ця інформація була зібрана за допомогою наступних кроків:

1. Для кожного з джерел було сформовано URL-адресу, яка вказує на список статей на цю тему.
2. Для кожного URL-адреси було здійснено запит до веб-сервера.
3. Для кожної отриманою сторінки було виконано парсинг HTML-коду.
4. Для кожної статті було отримано назву ресурсу, заголовок статті та текст статті.

В результаті було зібрано 407 статей, які були опубліковані з 24 лютого 2022 року до 15 березня 2023 року.

3.3.3 Збір текстів дописів природньої мови у датасет

Наступним етапом є акумулювання даних, які вимагають перевірки на наявність дезінформації. Для цього використовується парсер дописів (див. Додаток Б).

Зібрані дані походять із ресурсу “Twitter” із використанням хештегів, таких як #WarInUkraine, #StopWarInUkraine та інші. Структура датасету описується наступним чином:

Таблиця 3.3 – Опис зібраних полів дописів користувачів

Параметр	Опис
url	Посилання на твіт
date	Дата і час публікації твіту
content	Текст твіту
renderedContent	HTML-контент твіту
id	Унікальний ідентифікатор твіту
user	Користувач, який опублікував твіт
replyCount	Кількість відповідей на твіт
retweetCount	Кількість ретвітів твіту
likeCount	Кількість вподобань твіту
quoteCount	Кількість цитат твіту
conversationId	Унікальний ідентифікатор розмови, до якої належить твіт
sourceLabel	Джерело твіту (наприклад, веб-сайт, мобільний додаток)
outlinks	Посилання на сторонні веб-сайти, згадані в твіті
tcooutlinks	Посилання на скорочені посилання, згадані в твіті
media	Медіафайли, прикріплені до твіту (наприклад, фотографії, відео)
retweetedTweet	Чи є твіт ретвітом іншого твіту (значення 1 – ретвіт, 0 - оригінальний твіт)

Дані зберігаються у форматі CSV для зручності подальшого аналізу.

3.4 Реалізація бекенд-частини програмного продукту

У бекенді, реалізованому на Python, використовується потужна інфраструктура, що забезпечує високоефективну обробку та аналіз текстових даних, пов'язаних із війною в Україні, що дозволяє ефективно взаємодіяти з великим обсягом інформації та надає змогу отримувати надійні результати в реальному часі.

Схема взаємодії компонент відображена на Рисунку 3.1:

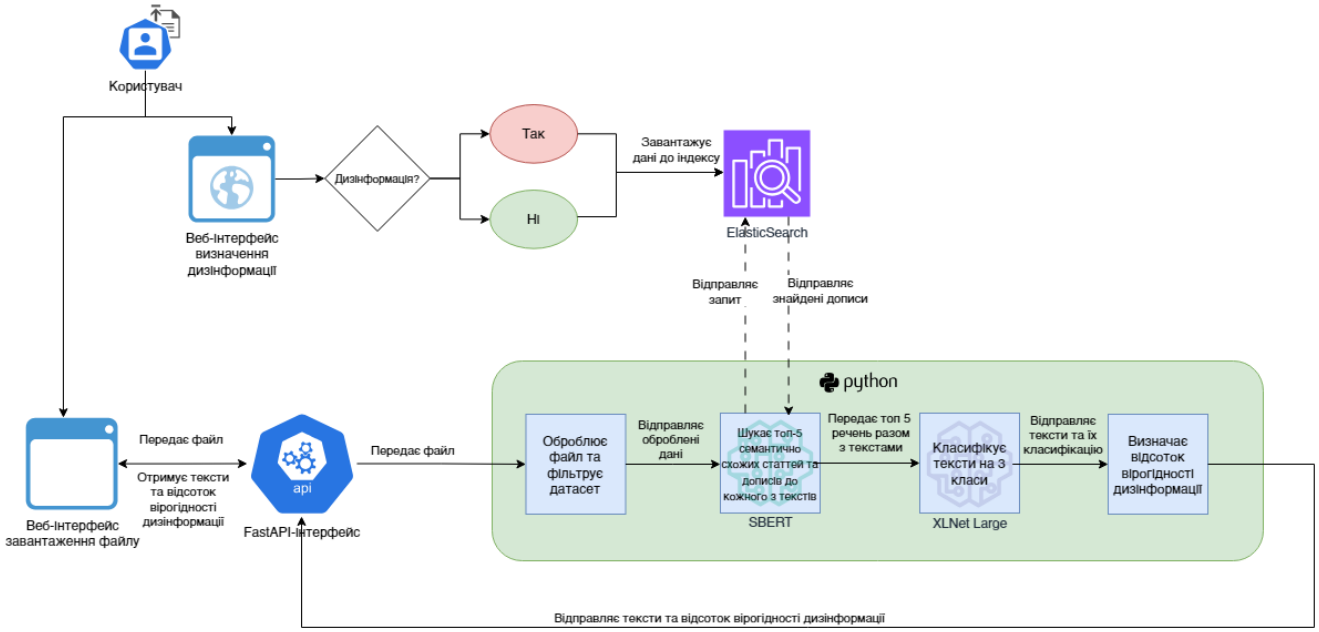

Рисунок 3.1 – Схема взаємодії компонент

Застосування модульного підходу дозволяє ефективно інтегрувати та оптимізувати роботу моделей розпізнавання фактів, спрямованих на виявлення інформації в текстах статей.

3.4.1 Реалізація передобробки тексту

Реалізована потужна система попередньої обробки текстових даних з метою підготовки їх для подальшого аналізу та моделювання. Початковий етап включає в себе функції для нормалізації тексту, видалення емодзі, обробки URL-адрес, а також для усунення зайвих символів та повторень.

Нормалізація тексту включає видалення ацентів, конвертацію унікоду в ASCII, транскодування сурогатних пар UTF-16 для емодзі, а також заміну певних токенів на стандартні позначення, наприклад, обов'язковий спецсимвол @.

Крім того, функції для роботи з емодзі дозволяють видаляти чи конвертувати їх у текст, забезпечуючи більш гнучкий підхід до обробки символів. Також реалізована можливість усунення надто довгих емодзі, що дозволяє уникнути проблем з індексами при подальшому аналізі.

Функції токенізації та нормалізації токенів включають обробку слів, які починаються з '@' або представляють собою URL-адреси. Це спрощує текст для подальшого аналізу та зменшує кількість унікальних токенів.

Усі ці етапи попередньої обробки здійснюються з урахуванням різноманітних особливостей текстових даних, що дозволяє створювати датасети, які можуть бути використані для різноманітних завдань аналізу та класифікації тексту.

3.4.2 Реалізація FastAPI

У цьому варіанті реалізації FastAPI відіграє важливу роль у визначенні веб-маршруту та обробці POST-запиту, який передає файл у форматі CSV. Система подальше взаємодіє з цим файлом безпосередньо в середовищі Python, проводячи необхідні операції обробки та аналізу. Цей механізм дозволяє ефективно і зручно взаємодіяти з вхідними даними, представленими у вигляді CSV, в процесі роботи з FastAPI.

Інтерфейс FastAPI Swagger відображений на Рисунку 3.2:

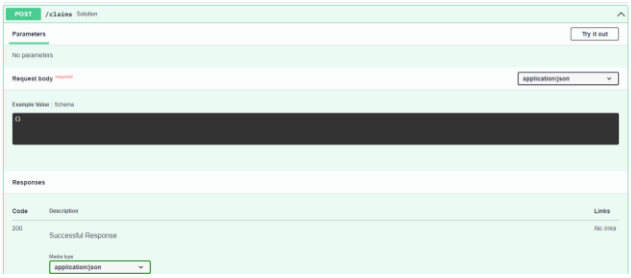

Рисунок 3.2 – Інтерфейс Swagger UI

Даний інтерфейс дозволяє відслідковувати та тестувати запити до API. Це є необхідним інструментом для налаштувань системи.

3.4.3 Реалізація ступеню визначення дизінформації у Python

У даному процесі аналізу документів використовується комплексний підхід, що об'єднує технології обробки природної мови (NLP) та машинного навчання для виявлення дезінформації. Початковий етап передбачає завантаження текстових даних з файлу формату CSV, які в подальшому піддаються обробці у Python.

Далі, отримані тексти піддаються аналізу за допомогою методів, що ґрунтуються на векторному представленні речень (SBERT) На наступному етапі використовується ElasticSearch Index для пошуку інформації. Це дозволяє виявити та відібрати топ-5 речень для кожного тексту. Використання SBERT спрощує розуміння схожості між реченнями, сприяючи точному визначенню ключових аспектів тексту.

Отримана інформація в подальшому піддається класифікації за допомогою моделі XLNet Large, що визначає ступінь узгодження між текстами (entailment), їхню нейтральність чи суперечливість (contradiction). Це дозволяє деталізувати рівень взаємодії між обраними реченнями. За допомогою методу majory voting визначаються найбільш ймовірні класифікації для кожного тексту серед трьох можливих класів. Цей підхід ґрунтується на принципі більшості серед найкращих варіантів, що допомагає зменшити можливі помилки та підвищити надійність результатів. Завершальний етап включає визначення ймовірності дезінформації, яка представлена у відсотках від 0 до 100. Цей показник створюється на основі агрегованих результатів попередніх етапів та відображає ступінь достовірності виявлення дезінформації в текстових даних.

3.4.4 Налаштування індексації у ElasticSearch

Вибір правильного типу даних для індексу – ключовий аспект. Необхідно визначити тип даних, який планується зберігати, і обрати відповідний тип поля для оптимального індексування та пошуку. Наприклад, для текстової інформації важливо використовувати тип “text”, щоб враховувати особливості аналізу тексту. Також є можливість використання аналізаторів, що допомагають оптимізувати обробку текстової інформації та забезпечують більш точний і контекстний пошук. Вигляд налаштування індексу у ElasticSearch відображено на Рисунку 3.3:

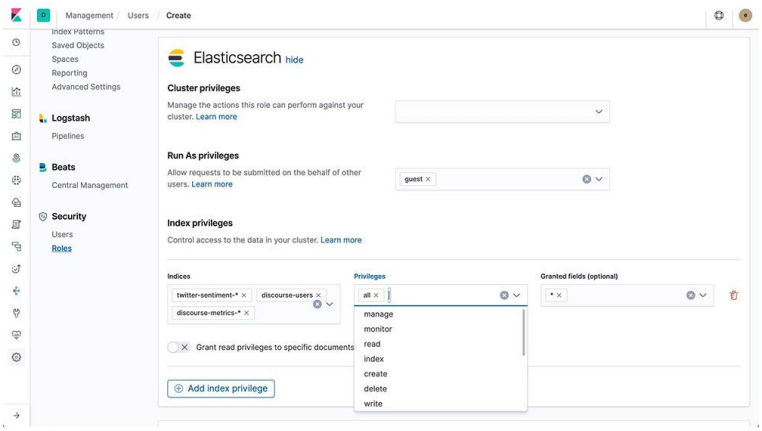

Рисунок 3.3 – Налаштування індексу у ElasticSearch

Важливим етапом є індексація лише необхідних полів, що дозволяє зменшити обсяг індексу та покращити продуктивність. Є можливість використання розподіленої архітектури та реплікації для забезпечення високої доступності та стійкості системи.

Враховуючи ці аспекти, налаштування ElasticSearch Index може значно покращити швидкість та продуктивність пошуку.

3.5 Реалізація фронтенд-частини програмного продукту

У даному програмному продукті, фронтенд реалізовано за принципом OODA. Принцип OODA (Observation, Orientation, Decision, Action) розроблений військовим стратегом і льотчиком Джоном Бойдом визначає ітеративний цикл прийняття рішень.

Перший етап – “Спостереження” – передбачає збір інформації про поточну ситуацію.

Другий етап – “Орієнтація” – включає аналіз зібраної інформації та оцінку можливостей, ґрунтуючись на знаннях, досвіді та культурному контексті. “Прийняття рішень”, третій етап, передбачає розробку стратегії або плану дій на основі попередніх спостережень та орієнтації.

Завершальний етап – “Дія” – включає виконання прийнятого рішення з метою зміни ситуації.

Цей цикл є ітеративним, з кожним новим циклом ґрунтуючись на отриманому досвіді та результаті попередніх етапів. Принцип OODA знаходить застосування в військовому та корпоративному менеджменті, а також в інших галузях, де ключовою є швидкість адаптації до змінюючихся умов.

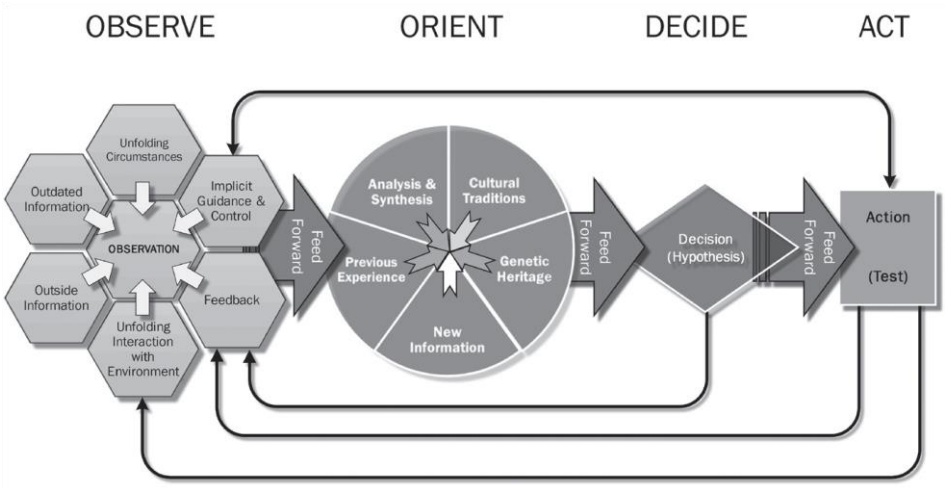

Рисунок 3.4 – Схема прийняття рішень OODA

Для забезпечення максимально ефективного досвіду використання кожного елементу платформи необхідно ретельно розглянути кожен етап принципу OODA [40]. Починаючи з етапу “Спостереження” (Observe), важливим є фіксування та збір інформації в узгодженому для платформи форматі перед початком роботи.

На етапі “Орієнтації” (Orient), після завантаження платформою необхідно забезпечити користувачеві чітку та дієву інформацію про дані, що враховує контекст та відповідає його потребам.

Етап “Прийняття рішень” (Decide) включає в себе використання результатів моделей ШІ для надання кінцевому користувачеві цілеспрямованої та пріоритетної інформації, що сприяє ефективному процесу прийняття рішень.

Накінець, на етапі “Дії” (Act) важливо надавати користувачеві можливість підтверджувати або відхиляти твердження, що підкріплені аналітикою. Окрім цього, система повинна надавати можливість внесення схвалених чи відхилених тверджень до бази знань, щоб забезпечити неперервний процес вдосконалення та розвитку.

Розроблене рішення несе значну користь для аналітика, забезпечуючи ефективність та оптимізацію управління завданнями. Однією з ключових переваг є економія часу, що стає можливою завдяки вдосконаленому інтерфейсу та використанню надійних моделей штучного інтелекту, що продуктивно розвиваються.

Оперативна сумісність стає невід'ємною перевагою, оскільки аналіз фактичних даних дозволяє здійснювати розрахунки ймовірності та покращує прийняття рішень. Це забезпечує глибокий і обґрунтований підхід до управління та стратегічного планування, забезпечуючи аналітика необхідними інструментами для оперативного реагування на зміни у ситуації.

Унікальна користь розробленого рішення полягає в його фокусі на ключових потребах користувачів, враховуючи потреби аналітика та забезпечуючи можливість розширення та вдосконалення. Рішення спрямоване на покращення ефективності управління, забезпечуючи необхідну гнучкість для адаптації до змін

у стратегічних завданнях та обставинах. Особливу перевагу для аналітика вносить функція масового завантаження даних, що дозволяє не лише економити час, але і отримувати дієві висновки. Це надає можливість проводити масштабний аналіз інформації, що в свою чергу сприяє прийняттю обґрунтованих та стратегічно значущих рішень в управлінській діяльності.

3.5.1 Інтерфейс користувача при вході у систему

Для того, щоб перейти на сторінку користувача, необхідно запустити код програмного продукту.

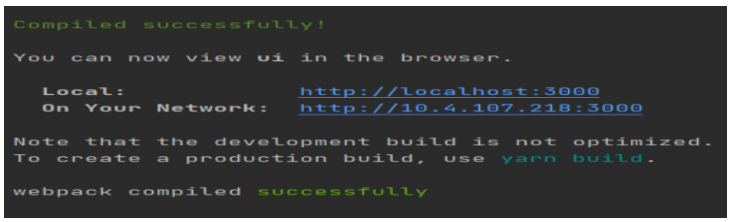

Рисунок 3.5 – Повідомлення про успішний запуск веб-застосунку

Після успішного запуску автоматично відкриється веб браузер, де користувач бачить початковий екран:

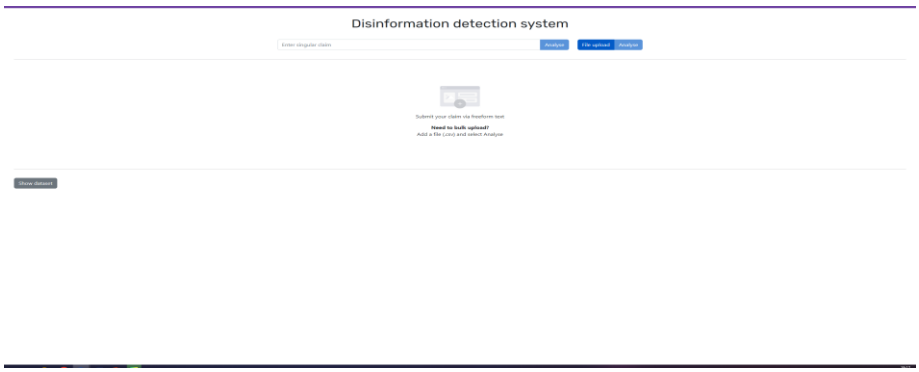

Рисунок 3.6 – Початковий екран користувача системи

Однією з ключових функцій є можливість завантаження датасету у форматі CSV, що відкриває широкі можливості для роботи з різноманітними даними. Ця опція дозволяє користувачу ефективно використовувати різні набори даних для аналізу та отримання висновків.

Додатково, інтерфейс включає опцію аналізу, яка дозволяє провести детальний розгляд поданої інформації на основі введених тверджень. Це створює зручний інструмент для користувача, який може систематично аналізувати та оцінювати введені дані.

Не менш важливим елементом є візуалізація даних на графіку, де відображено розподіл правдивих тверджень за часом та джерелами. Цей графік надає користувачеві об'єктивне уявлення про джерела інформації та їхню актуальність, допомагаючи визначити певні закономірності в наданні інформації.

3.5.2 Відображення графіків датасету програмного продукту

Основною особливістю є візуалізація даних на графіку. Графік відображає розподіл правдивих тверджень за часом та джерелами, зокрема “Редділт” та “Твіттер”. Для кожного твердження вказана дата його зібрання, а також кількість та джерело інформації. Ця візуалізація дозволяє користувачеві швидко та легко оцінити динаміку надходження правдивих тверджень залежно від часу та джерел.

Окрім того, графік також вказує на те, з яких конкретних датасетів складається підґрунтя правдивих тверджень.

Він розташований за кількістю та датою, надаючи користувачеві повний огляд над інформацією.

Кожен елемент графіка включає вказівку джерела, з якого було отримано відповідні дані.

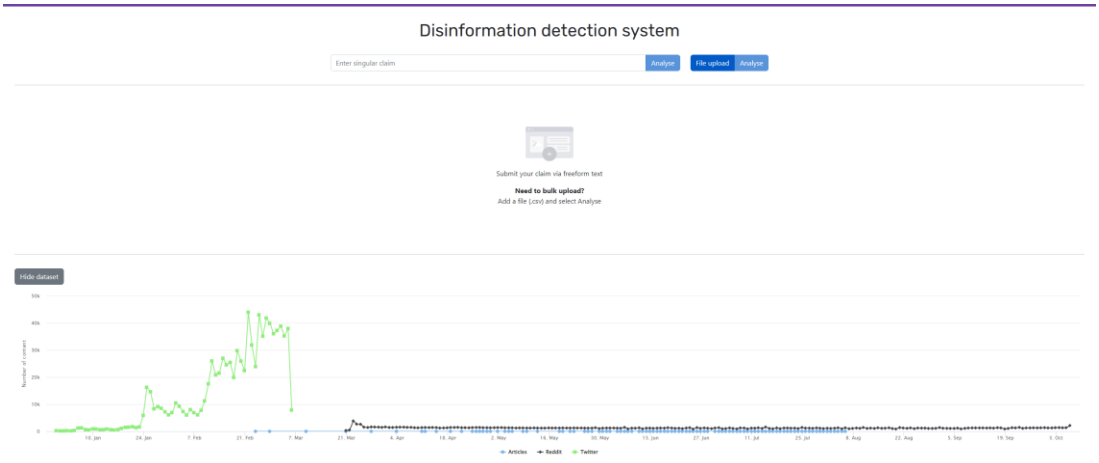

Рисунок 3.7 – Графік розподілу правдивих тверджень датасету

Такий підхід до візуалізації даних дозволяє користувачеві не лише аналізувати текстові дані, але і швидко отримувати об'єктивну картину розподілу правдивих тверджень за часом та джерелами, полегшуючи процес прийняття рішень на основі отриманих результатів.

3.5.3 Введення даних в текстове поле програмного продукту

Система має можливість взаємодії користувача з програмним інструментом, спрямованим на аналіз та визначення достовірності текстових тверджень. Введенням користувача є текстове твердження, яке стає об'єктом подальшого аналізу. Результати роботи програми подаються в формі визначення системою наявності або відсутності дезінформації у введеному твердженні.

Зазначеною функціональністю визначення достовірності текстових даних визнається ключовою властивістю системи. Користувач, відповідно, наділяється можливістю активної участі в процесі верифікації інформації, подаючи на розгляд програми власні твердження для аналізу.

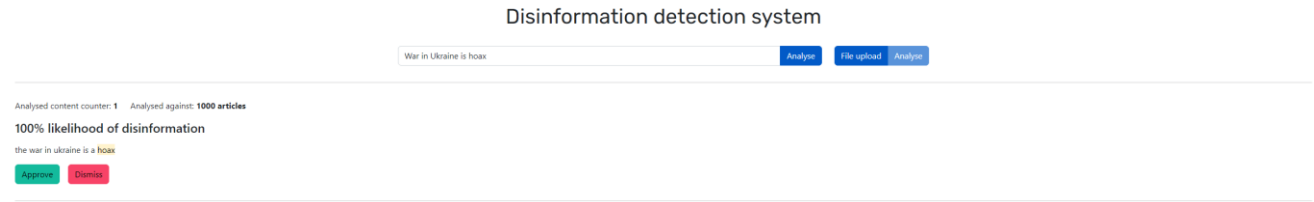

Рисунок 3.8 – Введення і відображення результатів роботи програмного продукту

Отримавши результати аналізу програмної системи, користувач, керуючись особистими переконаннями та рівнем наявної об'єктивної інформації, стає суб'єктом прийняття рішення щодо наявності дезінформації у введеному тексті. Цей етап визначається високим ступенем взаємодії користувача із здобутими результатами та підтримується особистим осудженням або підтвердженням обґрунтованості системно виявлених аномалій у текстових данних.

Інтерфейс програмного продукту, використовуючи концепцію OODA (спостереження, орієнтація, вирішення, дії), забезпечує користувачеві активну участь у процесі аналізу та прийняття рішення щодо правильності його функціонування.

Користувач має можливість здійснювати контроль та управління процесом верифікації інформації, що підвищує рівень довіри до системи.

Однією з ключових можливостей є можливість користувача підтверджувати чи відхиляти рішення, яке було прийнято системою.

Це відкриває додаткові можливості для вирішення ситуацій, де суб'єктивний досвід чи додаткові знання користувача можуть бути важливими для правильної інтерпретації результатів.

Після виклику дії через натискання, спостерігається анімаційний ефект, який служить для підтвердження користувацької взаємодії та сприяє її поліпшенню візуально.

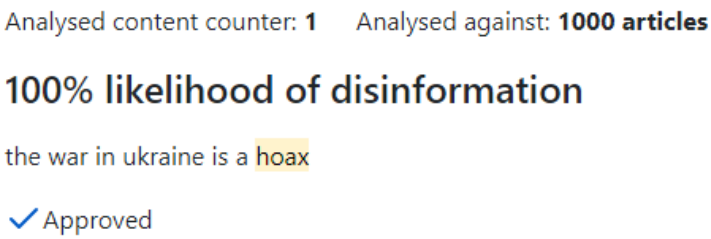

Рисунок 3.9 – Прийняття позитивного рішення користувачем

Після прийняття користувачем рішення відносно правильності роботи системи, здійснені дії реєструються в ElasticSearch. Ця дія включає в себе занесення тексту у відповідний індекс з урахуванням прийнятого рішення, де окремо виділяються індекси “groundtruth” та “hoax” для відзначення правдивих та неправдивих тверджень відповідно.

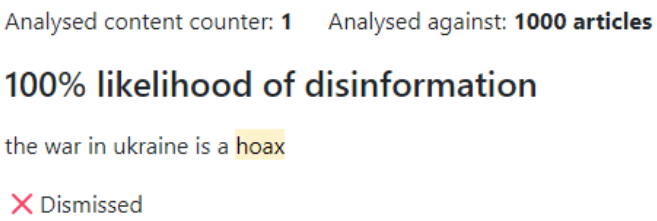

Рисунок 3.10 – Прийняття негативного рішення користувачем

Цей метод взаємодії не лише підвищує ефективність обміну інформацією між користувачем та системою, але й впроваджує систему збору даних та механізм звітування. Цей вбудований механізм аналізу дій користувача надає унікальну можливість отримання інсайтів, які в подальшому можуть бути використані для постійного удосконалення та оптимізації функціоналу програмного інструменту.

3.5.4 Введення даних через файл у інтерфейсі програмного продукту

Система дозволяє користувачеві взаємодіяти з програмним інструментом, призначеним для аналізу та визначення достовірності текстових тверджень, включаючи можливість завантаження файлу у форматі CSV.

Користувач може вводити як текстові твердження, так і завантажувати файли з набором тверджень для об'єктивного аналізу. Результати роботи програми виражаються у формі визначення системою наявності або відсутності дезінформації у введеному твердженні чи файлі.

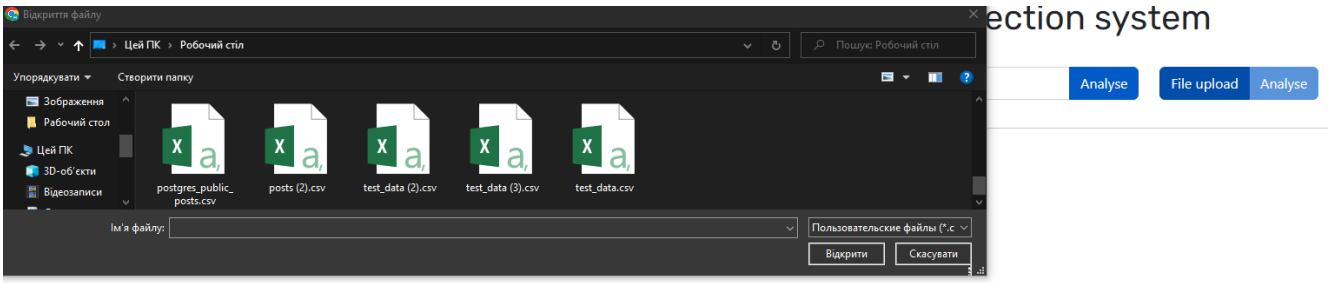

Рисунок 3.11 – Введення інформації з файлу CSV

Ця функціональність, яка охоплює обробку та аналіз файлів у форматі CSV, є важливою властивістю системи. Користувач має можливість активно брати участь у верифікації інформації, надаючи програмі не лише окремі текстові твердження, але й комплексні дані у вигляді файлу для більш ефективного аналізу.

Завантаживши дані, користувач може натиснути кнопку “Analyse”, щоб отримати результати роботи програми по виявленню дизінформації.

Процес аналізу та визначення наявності дезінформації у текстах користувача визначається натисканням кнопки “Analyse” в інтерфейсі програмного продукту. Ця дія спричиняє відображення усіх текстів, які були

вказані користувачем у файлі CSV, разом із відповідними вірогідностями наявності дезінформації.

Ключовий момент полягає в тому, що в результаті аналізу кожного текстового елементу, система виділяє вірогідність того, чи містить дане твердження дезінформацію. Це може бути візуалізовано на інтерфейсі, де для кожного тексту вказана відповідна міра впевненості чи ймовірності наявності дезінформації, що дозволяє користувачеві систематично оцінювати рівень довіри до кожного твердження.

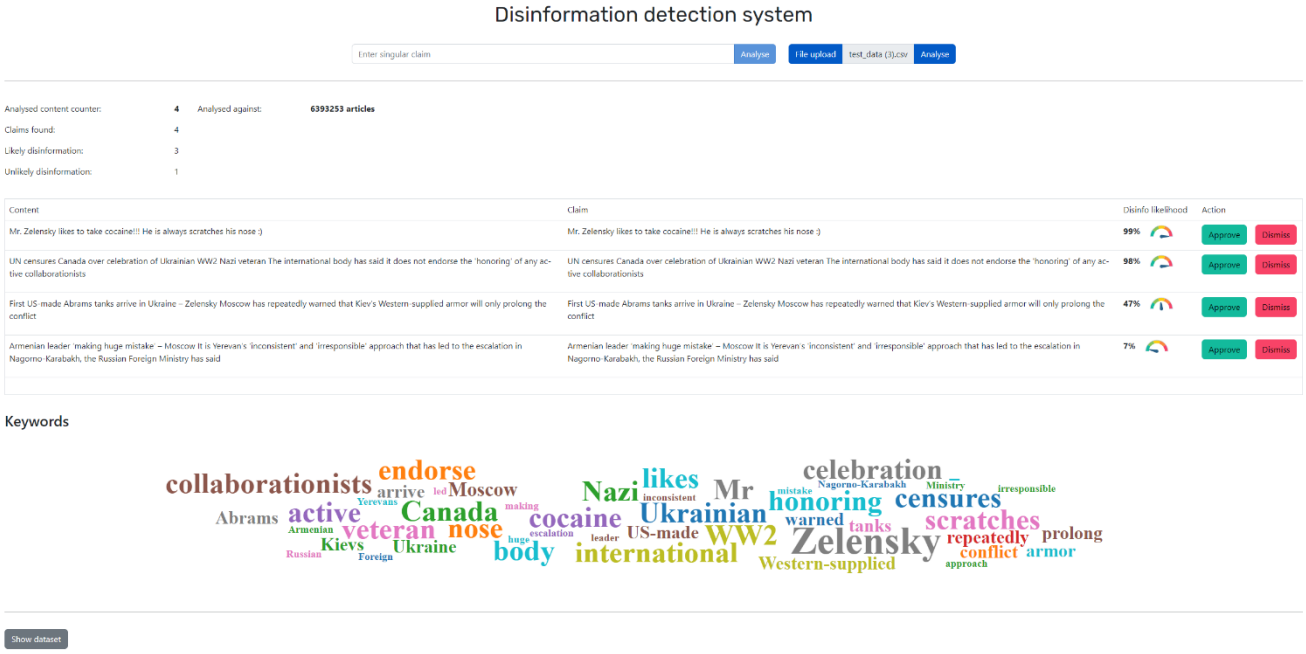

Рисунок 3.13 – Результат роботи системи по виявленню дизінформації з текстового файлу формату CSV

Такий підхід дозволяє користувачеві легко переглядати та аналізувати результати, інформуючись про рівень впевненості системи щодо наявності дезінформації у кожному конкретному тексті. Це створює ефективний та зручний інструмент для користувача, сприяючи об'єктивному оцінюванню та подальшому прийняттю рішень на основі результатів аналізу.

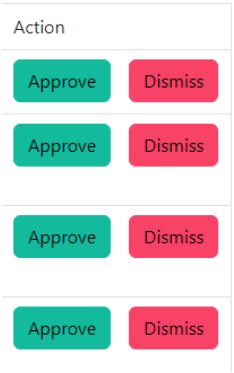

Рисунок 3.14 – Можливість прийняття рішення користувачем для кожного тексту

Графічне зображення “спідометра” надає інтуїтивно зрозумілу та візуально привабливу форму відображення рівня небезпеки дезінформації у кожному тексті. Відсоткове значення від 0 до 100 надає чіткий контекст, дозволяючи користувачеві оцінити ризик дезінформації на основі вірогідності, визначеної системою.

Ця інформація ще більше підсилюється розділенням спідометра на три кольорові зони – зелену, жовту і червону. Зелений колір вказує на низький ризик дезінформації, жовтий – на помірний, а червоний – на високий ризик. Такий колірний код додає додатковий рівень візуалізації та об'єктивності в оцінці рівня небезпеки для кожного тексту.

Таке розподілення на кольорові зони забезпечує користувачеві швидке та легке визначення рівня ризику для кожного тексту, надаючи ясний сигнал про ступінь достовірності інформації. Такий підхід допомагає забезпечити більш повне та комплексне сприйняття результатів аналізу для подальшого ухвалення обґрунтованих рішень стосовно наявності дезінформації у тексті.

Рисунок 3.15 – Графічне відображення «спідометра» з вірогідністю дезінформації

Після отримання відомостей про рівень ризику, користувач має можливість приймати рішення стосовно наявності дезінформації у кожному тексті. Це підкреслює активну участь користувача в процесі аналізу та визначення достовірності інформації, дозволяючи особисто оцінювати та приймати обґрунтовані рішення на основі результатів системного аналізу. Графічне представлення спідометра сприяє швидкому та чіткому сприйняттю інформації, полегшуючи користувачеві процес оцінювання та вирішення щодо наявності дезінформації у тексті.

3.5.5 Відображення ключових слів у інтерфейсі користувача

Додатковою важливою функцією програмного продукту є можливість відображення ключових слів, яке формується завдяки генерації word-cloud. Програма взаємодіє із вказаним CSV файлом, обирає всі речення та текстові дані з нього та використовує їх для створення визуально привабливого зображення ключових слів у формі word-cloud.

Word-cloud відображає найбільш значущі та часто зустрічаються слова у текстах, використаних для аналізу. Розмір та насиченість слова в word-cloud пропорційні їхній частоті та важливості в тексті. Таке графічне представлення

надає користувачеві можливість швидко визначити ключові терміни та теми, що є акцентованими у наданих текстових даних.

Рисунок 3.16 – Відображення ключових слів у інтерфейсі користувача

Цей функціонал надає користувачеві зручний інструмент для візуалізації та аналізу основних тем та ідей, які визначають зміст текстового матеріалу. Зображення word-cloud сприяє швидкому сприйняттю та розумінню суті тексту, що є важливим елементом для користувача під час загального ознайомлення з інформацією та орієнтації в ключових аспектах текстового контенту.

При натисканні на конкретне слово в word-cloud, програма реалізує функціонал відображення всіх текстів з файлу CSV, які містять вказане ключове слово, разом із зазначенням рівня дезінформації. Це дозволяє користувачеві отримати більш детальний та контекстуальний погляд на вміст текстових даних, пов'язаних із обраною ключовою темою чи терміном.

3.5.6 Підсумок роботи аналітика по виявленню дезінформацію

В заключному етапі аналізу, користувач може скористатися функціоналом кнопки "Analyst Outcome", щоб переглянути аналітичну звітність щодо проведеного аналізу. На даному екрані відображається інформація, яку враховував аналітик під час своєї діяльності.

Зокрема, в звіті доступний контент, який був підданий аналізу аналітика, включаючи тексти, що визначалися у вказаному CSV файлі. Крім того, в звіті

відображається вірогідність дезінформації, яку система визначила під час автоматичного аналізу.

Найважливішою частиною аналітичної звітності є рішення аналітика, де вказується, чи підтверджено або відхилено визначену системою вірогідність дезінформації. Також вказується дата, коли аналітик прийняв дане рішення.

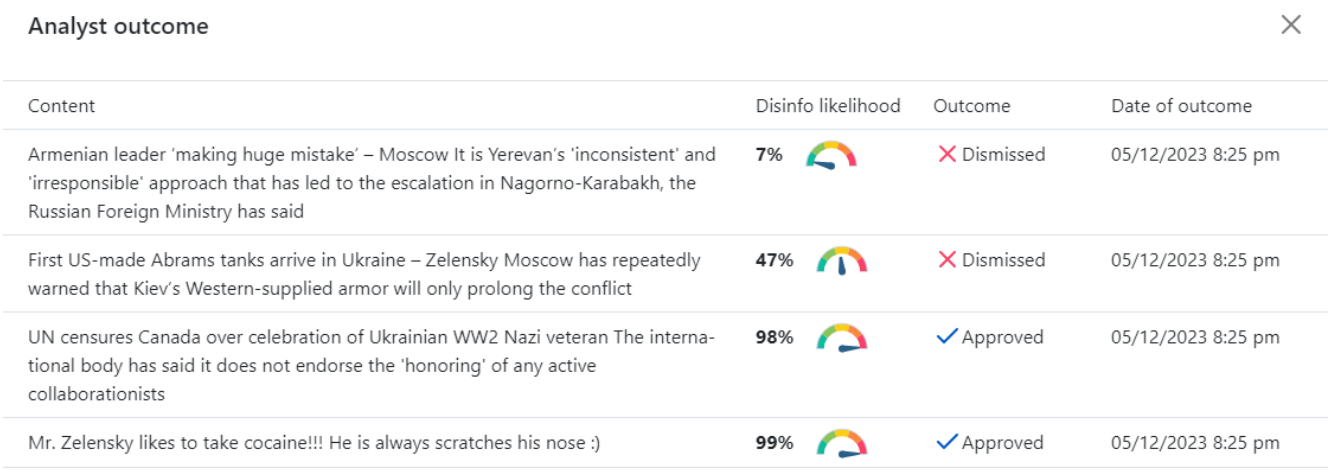

Рисунок 3.18 – Результат роботи аналітика

Цей функціонал забезпечує користувача із загальним підсумком роботи аналітика, дозволяючи відстежувати процес аналізу та прийняття рішень. Такий підхід допомагає забезпечити прозорість у внутрішніх процесах та діях аналітичного робочого потоку. Також даних підхід точно підходить для принципу OODA.

Висновки до розділу 3

У другому розділі було розглянуто програмну реалізацію методу виявлення дезінформації в текстах ЗМІ.

В ході розробки було проведений аналіз та вибір оптимальних інструментів для реалізації фронтенду та бекенду. На бекенді обрано мову програмування Python у поєднанні з фреймворком FastAPI та моделями нейронних мереж SBERT та XLNet Large.

Щодо фронтенду, використовуються технології JavaScript та React. React надає зручний інтерфейс для розробки динамічних веб-сторінок, а JavaScript є основною мовою програмування для взаємодії з користувачем на стороні клієнта.

Цей стек технологій дозволяє забезпечити високу швидкість та ефективність роботи системи, а також ефективну обробку текстових даних, що є ключовим аспектом у проекті, орієнтованому на аналіз та визначення дезінформації. Підходом до створення зручного інтерфейсу слугував підхід OODA.

Після було розглянуто основний сценарій користувача системи із зазначенням кожної можливості використання.

4 ОБЧИСЛЮВАЛЬНІ ЕКСПЕРИМЕНТИ МОДЕЛЕЙ ОБРОБКИ ПРИРОДНЬОГО ТЕКСТУ

У четвертому розділі було проведено обчислювальні експерименти для вибору найкращої моделі класифікації тексту для побудови архітектури визначення дезінформації. Розділ містить порівняння трьох моделей для класифікації схожості тексту та дві моделі для класифікації дезінформації.

4.1 Порівняння точності моделей схожості тексту сімейства BERT

Модель "RoBERTa-Base" відзначається використанням нових методів оптимізації, що дозволяє моделі швидше навчатися, що робить її особливою у визначенні швидкості визначення дезінформації. Далі, "all-mpnet-base-v2" виявляє використання архітектури Transformer-XL, яка дозволяє моделі краще розуміти довгі речення та тексти, що відрізняють її від інших моделей. Нарешті, "All-MiniLM-L6-v2" проявляє швидкість роботи, завдяки мінімальному розміру параметрів, що робить її більш ефективною і вимагає менше ресурсів для навчання та використання, які варто врахувати при порівнянні з іншими моделями.

Ці результати можуть визначати найбільш відповідні моделі для конкретних завдань або вказувати на потребу додаткового дослідження для визначення оптимального вибору в конкретному в контексті використання моделей для обробки природної мови. Проведення тестування відбувалось на трьох датасетах: SQuAD 2.0, RACE та вже існуючому датасеті Twitter і The Guardian та New York Times.

4.1.1 Оцінка якості класифікації моделі RoBERTa-base

У даному дослідженні модель RoBERTa-base була піддана тестуванню на трьох різних наборах даних з метою оцінки її ефективності у різних завданнях.

SQuAD 2.0 використовується для оцінки здатності моделі відповідати на запитання, засновані на фактичних текстах. В тестовій вибірці з 100 елементів досягнута точність на рівні 96.3%, що свідчить про високий рівень успішності в розв'язанні завдань на визначення відповідей у контексті структурованих текстів.

Набір даних RACE використовується для вимірювання здатності моделі розуміти та аналізувати прочитаний текст. З точністю 92.1% на тестовій вибірці з 100 елементів, модель RoBERTa-base продемонструвала високу ефективність в завданнях розуміння прочитаного тексту.

Для оцінки роботи моделі в умовах реальних текстів, тестова вибірка, складена з 100 твітів та 407 дописів з The Guardian і The New York Times, була взята. Результати показали точність на рівні 74.0%, що вказує на здатність моделі до адекватного розуміння та обробки текстів із соціальних мереж та новинарських платформ.

Загальною висновком є те, що модель RoBERTa-base успішно впоралася з різноманітними завданнями на різних наборах даних, що підтверджує її високий рівень адаптабельності та ефективності у визначених контекстах використання.

Протестувавши на кожному з датасетів маємо результат роботи класифікації. Точність визначалась за метрикою f1-score (4.1):

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{4.1}$$

де Precision (точність) – це відношення кількості правильно визначених позитивних класів (True positives) до загальної кількості екземплярів, які були класифіковані як позитивні (true positives + false positives):

$$Precision = \frac{True\ positives}{True\ positives + False\ positives} \quad (4.2)$$

і Recall (повнота) – це відношення кількості правильно визначених позитивних класів (true positives) до загальної кількості екземплярів, які фактично є позитивними (true positives + false negatives). Формула для повноти:

$$Precision = \frac{True\ positives}{True\ positives + False\ negatives} \quad (4.3)$$

F1-score використовується для оцінки моделей класифікації, збалансовуючи між точністю і повнотою. Високий F1-score свідчить про те, що модель має як високу точність, так і високу повноту.

Результати роботи класифікації відображно на рисунку 4.1:

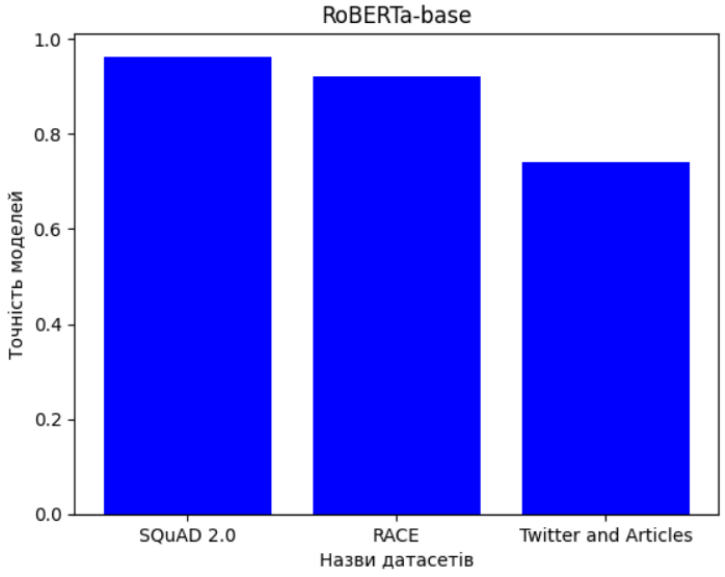

Рисунок 4.1 – Результати роботи моделі RoBERTa-base

В результаті роботи маємо точності для кожного з датасетів. Як видно з графіків, дані, які були зібрані для визначення дезінформації мають найменшу

точність. Може бути декілька причин: особливість природи даних, людський фактор (оскільки обробка природньої мови), недостатня очистка даних.

4.1.2 Оцінка якості класифікації моделі All-MiniLM-L6-v2

У продовженні дослідження моделей, необхідно перевірити точність All-MiniLM-L6-v2. Ця модель в свою чергу також була піддана тестуванню на трьох різних наборах даних з метою оцінки її ефективності у різних завданнях.

Використовувались аналогічні датасети для тестування роботи моделі, а також F1-Score (4.1). В результаті маємо:

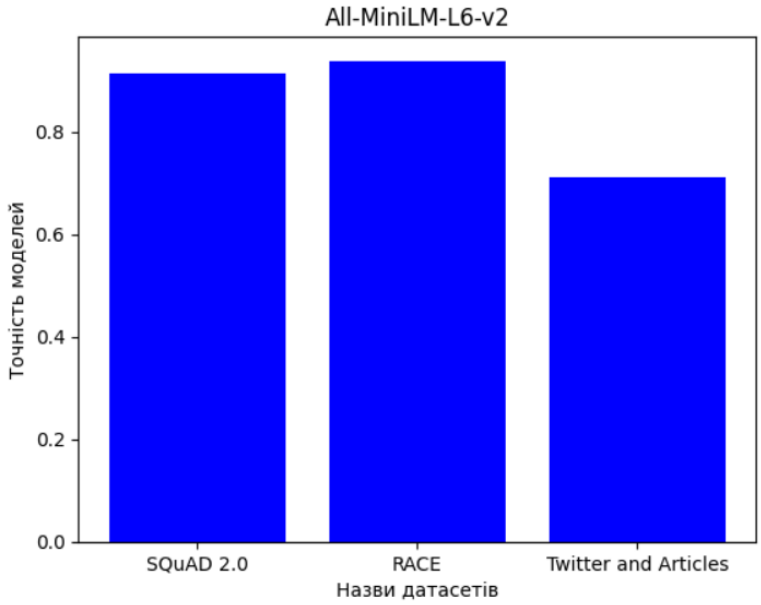

Рисунок 4.2 – Результати роботи моделі All-MiniLM-L6-v2

Результат роботи з датасетом SQuAD 2.0 становить 91.3% точності, в свою чергу RACE складає 93.8%. І з датасетом Twitter and Articles результат складає 71.1%.

Результати дещо різняться у порівнянні з RoBERTa-base. Причиною цього може слугувати здатність моделі до аналізу повідомлень невеликого розміру, які урізаються. За рахунок цього в датасеті RACE точність більша, ніж у SQuAD 2.0, однак попередня модель має результати навпаки. Точність у датасеті Twitter and Articles дещо зменшилась.

4.1.3 Оцінка якості класифікації моделі all-mpnet-base-v2

Модель all-mpnet-base-v2 була піддана тестуванню на трьох наборах даних з метою визначення ефективності порівняння двох синтаксичних одиниць. Результат роботи класифікації відображено на рисунку 4.3

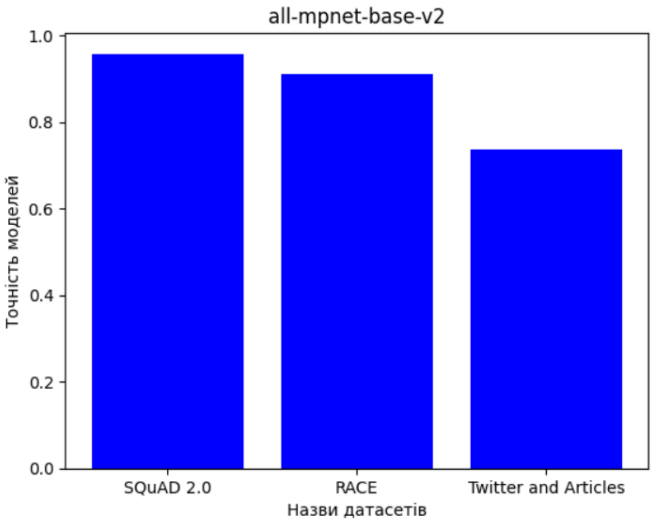

Рисунок 4.3 – Результати роботи моделі all-mpnet-base-v2

Результат роботи з датасетом SQuAD 2.0 становить 95.8% точності, в свою чергу RACE складає 91.2%. І з датасетом Twitter and Articles результат складає 73.8%.

Однак повторно модель видає інші результати, які деяк варіюються між собою. Це зумовлено випадковими перестановками в моделі, які можуть змінювати результат.

4.2 Аналіз результатів та вибір моделі порівняння схожості тексту

Порівнюючи результати трьох моделей, RoBERTa-base, All-MiniLM-L6-v2 та all-mpnet-base-v2, відзначається, що кожна з них має свої переваги та недоліки. RoBERTa-base відзначається високою точністю на вхідних даних, але при цьому виявляє меншу ефективність у відношенні часу роботи порівняно з іншими розглянутими моделями.

З іншого боку, all-mpnet-base-v2 може виявити високу продуктивність, проте її значною недолікою є великий час, необхідний для процесу навчання.

All-MiniLM-L6-v2, хоч і не досягає найвищого рівня точності на даних, відрізняється найвищою швидкістю виконання серед розглянутих моделей. Це стає перевагою, зокрема, через здатність суттєво зменшити час рантайму.

Крім того, модель виявляється більш ефективною у процесі навчання, що може бути важливим аспектом при опрацюванні великого обсягу даних.

Зазначимо також, що вона вирізняється простотою використання та налаштування, що робить її зручною для користувачів, які високо цінують ефективність та простоту взаємодії з моделлю.

Крім того, слід зауважити, що RoBERTa-base вирізняється високим рівнем роботи з мовною моделлю, зокрема у розпізнаванні семантичних зв'язків та взаємодії з контекстом. З іншого боку, All-MiniLM-L6-v2, завдяки своїй легкості та швидкості, може бути особливо корисним у завданнях, де обробка текстової інформації в режимі реального часу є важливою.

4.3 Порівняння точності моделей класифікації тексту

4.3.1 Оцінка якості класифікації моделі xlnet-large-cased

У ході даного дослідження модель xlnet-large-cased була протестована на трьох різних наборах даних для оцінки її ефективності в різноманітних завданнях.

SQuAD 2.0 використовувався для оцінки здатності моделі відповідати на запитання, засновані на фактичних текстах.

У тестовій вибірці з 100 елементів досягнута точність на рівні 87%, що свідчить про високий рівень успішності у вирішенні завдань з визначення відповідей в контексті структурованих текстів.

Набір даних RACE використовувався для вимірювання здатності моделі розуміти та аналізувати прочитаний текст. З точністю 82% на тестовій вибірці з 100 елементів, модель xlnet-large-cased продемонструвала високу ефективність в завданнях розуміння прочитаного тексту.

Для оцінки роботи моделі в умовах реальних текстів, тестова вибірка, що складалася з 100 твітів та 407 дописів з The Guardian і The New York Times, була використана.

Результати показали точність на рівні 72%, що свідчить про здатність моделі до адекватного розуміння та обробки текстів із соціальних мереж та новинарських платформ.

F1-Score, який використовується у нашому дослідженні для оцінки моделей класифікації текстів, є ключовим показником, що балансує між точністю і повнотою. Цей метричний показник стає особливо важливим у випадках, коли обидві ці характеристики є критичними для ефективності моделі.

Високий F1-Score свідчить про те, що модель досягає не лише високого рівня точності в правильному класифікаційному процесі, але й високої повноти виявлення об'єктів цікавості. Отже, моделі, які мають високий F1-Score, можуть ефективно працювати з різноманітними типами текстів та розпізнавати дезінформацію навіть у складних випадках.

Використання F1-Score у нашому дослідженні дозволяє нам об'єктивно оцінити не лише точність наших моделей, але й їхню здатність ефективно виявляти дезінформацію в текстових джерелах. Цей підхід до оцінки дозволяє враховувати як фактичні помилки в класифікації, так і пропущені можливість виявлення об'єктів цікавості, забезпечуючи комплексний погляд на ефективність наших моделей.

Результати роботи класифікації відображно на рисунку 4.4:

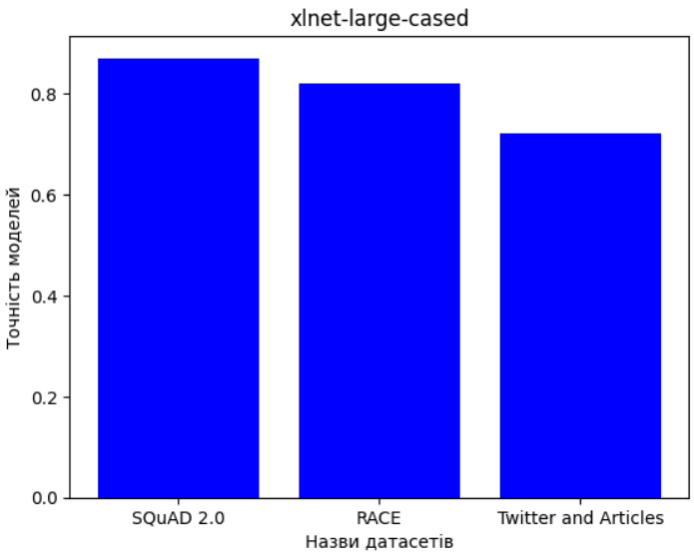

Рисунок 4.4 – Результати роботи моделі xlnet-large-cased

Після виконання аналізу було отримано результати точності для кожного з наборів даних. З діаграм видно, що дані, які були зібрані для визначення дезінформації, відзначаються найменшою точністю. Це може бути пояснено декількома можливими причинами, такими як особливості природи даних, вплив людського фактору (оскільки використовується обробка природньої мови) або недостатня очистка даних.

4.3.2 Модель bert-base-uncased

Модель bert-base-uncased була піддана тестуванню на трьох наборах даних з метою визначення ефективності класифікацію тексту. Результат роботи класифікації відображено на рисунку 4.5.

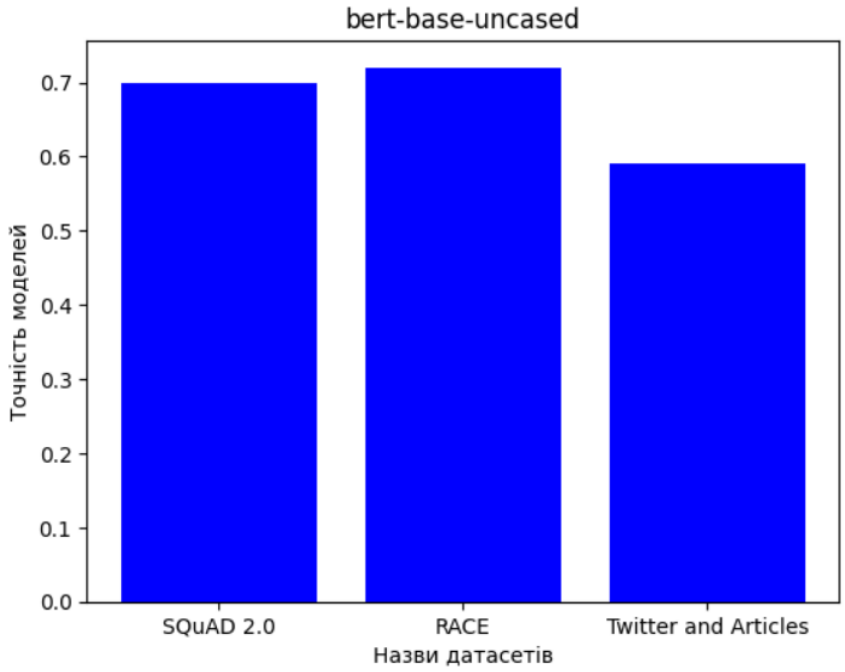

Рисунок 4.5 – Результати роботи моделі bert-base-uncased

В цілому, можна відзначити, що загальна точність моделі BERT-base-uncased є меншою у порівнянні з іншими моделями, при цьому спостерігається більша тривалість часу обробки.

4.4 Аналіз результатів та вибір моделі класифікації тексту

Обидві моделі, xlnet-large-cased та bert-base-uncased, мають свої власні переваги та недоліки. Однак, в ході порівняльного аналізу, виявлено, що xlnet-

large-cased проявила кращі результати у завданнях класифікації порівняно з bert-base-uncased. Зокрема, вона виявилася більш ефективною у вирішенні завдань, пов'язаних із визначенням класів або категорій текстів.

Зазначимо, що xlnet-large-cased також відзначається високою продуктивністю та швидкістю роботи порівняно з bert-base-uncased. Це може бути особливо важливим фактором при обробці великих обсягів даних або в умовах обмежених ресурсів.

4.5 Архітектура роботи ансамблю моделей порівняння схожості тексту та класифікації

У нашому підході до визначення схожих текстів використовується модель All-MiniLM-L6-v2 для визначення топ-5 найбільш схожих текстів з заданого тексту. Після отримання цих п'яти текстів ми подається вихідне речення або текст разом з цими п'ятьма текстами на вхід моделі XLNet-large-cased.

Модель XLNet-large-cased порівнює кожен з цих текстів з вихідним текстом.

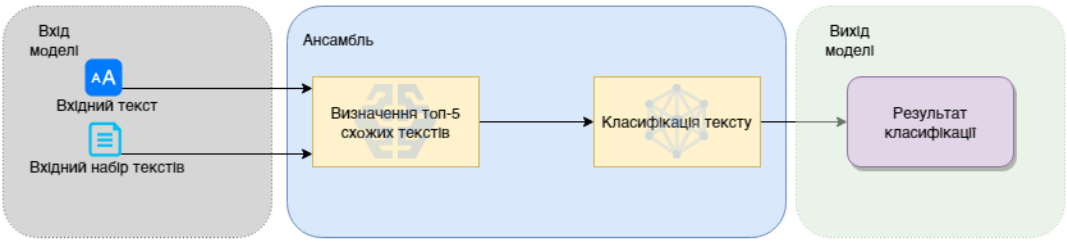

Рисунок 4.6 – Архітектура ансамблю моделей

Порівняльні таблиці точностей для кожної з архітектур наведено нижче (таблиці 4.1, 4.2, 4.3).

Таблиця 4.1 – Порівняння точностей моделей за тестовою вибіркою SQuAD

Перша модель архітектури	**Друга модель архітектури**		**Точність, f-score**
	xlnet-large-cased	bert-base-uncased	
RoBERTa-base	✓		0.76
		✓	0.65
all-mpnet-base-v2	✓		0.77
		✓	0.6
All-MiniLM-L6-v2	✓		0.74
		✓	0.54

Таблиця 4.2 – Порівняння точностей моделей за тестовою вибіркою RACE

Перша модель архітектури	**Друга модель архітектури**		**Точність, f-score**
	xlnet-large-cased	bert-base-uncased	
RoBERTa-base	✓		0.79
		✓	0.66
all-mpnet-base-v2	✓		0.79
		✓	0.62
All-MiniLM-L6-v2	✓		0.73
		✓	0.55

Таблиця 4.3 – Порівняння точностей моделей за тестовою вибіркою Tweets and Articles

Перша модель архітектури	**Друга модель архітектури**		**Точність, f-score**
	xlnet-large-cased	bert-base-uncased	
RoBERTa-base	✓		0.71
		✓	0.54
all-mpnet-base-v2	✓		0.72
		✓	0.55
All-MiniLM-L6-v2	✓		0.69
		✓	0.43

Висновки до розділу 4

У четвертому розділі нашого дослідження ми вивчили та аналізували різноманітні моделі з метою визначення оптимальних підходів до класифікації текстів. Зокрема, результати нашого аналізу вказали на високу ефективність двох конкретних моделей - All-MiniLM-L6-v2 та xlnet-large-cased.

Модель All-MiniLM-L6-v2 вражає своєю винятковою швидкодією та ефективністю. Використання мінімалізованої мовної моделі шостого рівня дозволило досягти високої точності при виявленні дезінформації в текстових матеріалах. Її здатність працювати із скороченими варіантами мови та швидка реакція на різноманітні лінгвістичні виклики роблять її важливим інструментом у контексті аналізу текстів ЗМІ та соціальних мереж.

Друга високоефективна модель, xlnet-large-cased, використовує трансформерну архітектуру, яка дозволяє їй аналізувати контекст тексту та взаємодіяти зі словами в оберненому порядку. Ця особливість надає моделі додаткові можливості у розумінні глибокого змісту текстів, що робить її ефективною для виявлення дезінформації.

З метою подальшого покращення точності визначення дезінформації, ми вирішили використовувати ансамбль обох моделей. Цей підхід дозволяє поєднати сильні сторони кожної моделі та уникнути їхніх можливих обмежень, створюючи високоефективний інструмент для виявлення дезінформації в різноманітних текстових матеріалах, незалежно від джерела.

5 РОЗРОБКА СТАРТАП-ПРОЄКТУ

У п'ятому розділі наведено опис ідеї стартап проєкту, визначено його технічний аудит, проаналізовано ринкові можливості та визначено базову стратегію.

5.1 Опис ідеї

У таблиці 5.1 приведено загальну ідею стартап-проєкту:

Таблиця 5.1 – Опис ідеї стартап-проєкту

Зміст ідеї	**Застосування**	**Вигода для користувача**
Виявлення дезінформації в текстах	Масова інформація (новини, соціальні мережі), коментарі, корпоративні текстові дані	Підвищення рівня достовірності інформації, захист від маніпуляцій та обману

Після визначення стартап-проєкту, необхідно провести детальний аналіз конкурентів, оцінюючи як їхні переваги, так і недоліки. Важливо ретельно вивчити конкурентне середовище, звертаючи увагу на тільки не тільки сильні сторони, але й слабкі сторони конкурентів. Під час аналізу слід розглядати можливості для вдосконалення власного продукту або послуги на основі виявлених недоліків у конкурентів.

Таблиця 5.2 – Аналіз продукту та конкурентів

№ п/п	Економічні характеристики ідеї	Потенційні концепції конкурентів			W	N	S
		Дипломний проект	FactCheck	Snopes			
1	Витрати на розробку	Витрати на розвиток алгоритмів машинного навчання та інтеграцію	Великі втрати на вахівців та перевірку фактів	Спеціалізовані розслідування та ресурси для факт-чеку			✓
2	Витрати на просування	Рекламні кампанії та партнерства для популяризації інструменту виявлення дезінформації.	Велика аудиторія через партнерські угоди з медіа	Специфічний контент та співпраця з медійними платформами	✓		
3	Цінова стратегія та модель взаємодії з користувачами	Безкоштовний базовий доступ з можливістю покупки розширених функцій.	Безкоштовний доступ для користувачів, підтримка від благодійних фондів	Платні підписки для додаткових функцій та контенту			✓
4	Реклама та розвиток партнерств	Партнерства з медійними платформами для підвищення обізнаності проекту.	Партнерства з видавництвами та соціальними мережами	Рекламні партнерства та співпраця з образовчими закладами.		✓	

5.2 Технологічний аудит проекту

Далі необхідно проаналізувати технологічну здійсненість проєкту. Детальний аналіз наведено у таблиці 5.3:

Таблиця 5.3 – Технологічна здійсненість проєкту

№ п/п	Ідея проєкту	Технології реалізації	Наявність технологій	Доступність технологій
1	Виявлення дезінформації в текстах	Мова програмування Python	Наявні	Доступні
2		Мова програмування C++	Відсутні	Недоступні
3		Мова програмування Java	Наявні	Недоступні

Таким чином було обрано мову програмування Python.

5.3 Аналіз ринкових можливостей

Щоб побудувати вдалий стартап, необхідно проаналізувати ринок стартапу, на який він претендує. Таблиця 5.4. Має чіткий опис ринку можливостей.

Таблиця 5.4 – Аналіз ринку стартап-проєкту

№ п/п	Показники ринку	Характеристика
1	Кількість конкурентів	~5
2	Обсяг продажів, од.	5.5 млн. $
3	Динаміка ринку	Стрімко зростає
4	Середня норма рентабельності (%)	15

В результаті ринок є найбільш зростаючим та найбільш привабливим для входження, оскільки має невелику кількість конкурентів та має багато потенційних користувачів.

Таблиця 5.5 – Аналіз цільової аудиторії стартап-проєкту

Потреба, що формує ринок	**Цільова аудиторія**	**Відмінність у поведінці потенційних груп клієнтів**	**Вимоги споживачів до товару**
Виявлення дезінформації в текстах засобах масової інформації і дописах	Медійні платформи: Новинні портали, соціальні мережі. Корпорації: Компанії, які використовують масові медіа для комунікації. Звичайні користувачі: Люди, які активно споживають інформацію в Інтернеті.	Медійні платформи: Редактори, журналісти, редакційні бюро. Корпорації: Вищі керівники, корпоративні комунікації. Звичайні користувачі: Активні Інтернет-користувачі, освічені споживачі інформації.	Медійні платформи: Велика кількість публікацій, акцент на швидкості поширення новин. Корпорації: Фокус на управлінні репутацією та внутрішній інформації. Звичайні користувачі: Активна участь у соціальних мережах, уважне сприйняття новин.

Процес обчислення фактору загроз (див. таблицю 5.6) важливий для визначення потенційних викликів та обмежень, які можуть виникнути при введенні продукту на ринок. Цей аналіз сприяє розробці стратегій мінімізації ризиків та підготовці до можливих негативних сценаріїв.

У паралельному розгляді фактори можливостей (зазначені в таблиці 5.7) виконують важливу роль у визначенні переваг та перспектив для продукту на ринковому середовищі. Ці фактори служать основою для розрахунку конкретних можливостей, які можуть бути використані для досягнення успішного впровадження та подальшого розвитку продукту.

Таблиця 5.6 – Фактори загроз

Фактор загрози	**Зміст загрози**	**Можлива реакція компанії**
Технічні атаки на систему	Спроби зламу або атаки на інформаційні системи компанії для отримання несанкціонованого доступу або завадження роботі.	Розвиток та вдосконалення кіберзахисту, шифрування даних, резервне копіювання.
Репутаційні атаки	Напади на репутацію компанії, в тому числі розповсюдження дезінформації про продукт або послуги.	Активна комунікація з громадськістю, виявлення та відповідь на дезінформацію.
Юридичні та регуляторні ризики	Зміни в законодавстві або внесення нових нормативів, які можуть вплинути на діяльність компанії.	Співпраця з юридичними експертами, відстеження змін в законодавстві, відповідь на вимоги.
Конкурентні атаки	Дії конкурентів, спрямовані на виведення компанії з ринку або дискредитацію її продуктів та послуг.	Постійний моніторинг конкурентів, стратегічне позиціонування, інновації та реклама.
Соціальні та культурні фактори	Вплив соціальних або культурних тенденцій на сприйняття продукту або послуг компанією.	Адаптація маркетингових стратегій, вивчення та врахування потреб різних соціальних груп.
Економічні ризики	Зміни в економічних умовах, такі як фінансова криза, що може вплинути на фінансовий стан компанії.	Диверсифікація ресурсів, стабілізація фінансового стану, аналіз економічних тенденцій.

Таблиця 5.7 – Фактори можливостей

Фактор можливості	**Зміст можливості**	**Можлива реакція компанії**
Технологічні інновації	Завдяки новим технологіям можливі вдосконалення продукту, швидший розвиток та конкурентні переваги.	Інвестування в дослідження та розробку, адаптація та інтеграція нових технологій.

Кінець таблиці 5.7.

Збільшення попиту на ринку	Зростання інтересу аудиторії до продукту чи послуги, можливість залучення нових клієнтів.	Розширення маркетингових кампаній, підтримка росту виробництва та обслуговування клієнтів.
Регуляторні підтримка та зміни	Підтримка з боку регуляторів чи зміни в законодавстві, які можуть вплинути на ринкові умови.	Вивчення та використання нових правил, співпраця з регуляторами, адаптація до змін.
Збільшення соціального впливу	Можливість створення позитивного соціального впливу, що призводить до підвищення репутації компанії.	Впровадження соціальних ініціатив, комунікація про соціальний вплив, співпраця з громадськістю.
Глобальний ринок	Можливість розширення на нові географічні ринки та отримання доступу до нових клієнтів та можливостей.	Розробка глобальної маркетингової стратегії, адаптація до місцевих особливостей ринків.
Партнерства та стратегічні зв'язки	Встановлення партнерств та стратегічних зв'язків для спільного розвитку та взаємовигідної співпраці.	Активний пошук потенційних партнерів, розробка та управління стратегічними партнерствами.

Далі необхідно провести аналіз пропозиції, в ході якого визначаться риси конкуренції для даного стартап-проєкту. Це дозволить в подальшому визначити та обрати стратегію ринку.

Для цього будується таблиця зі ступеневим аналізом ринку конкурентів, яка показує фактори, можливості та реакції компанії.

Таблиця 5.8 – Ступеневий аналіз ринку конкурентів

Фактор ринкового конкуренту	**Зміст можливості**	**Можлива реакція компанії**
Тип конкуренції: Олігополія	Змагання за інновації, нові технології для поліпшення продукту або зменшення витрат.	Інвестування в дослідження та розробку, пошук інновацій та новаторських рішень.
За рівнем конкурентної боротьби: міжнаціональна	Боротьба за увагу споживачів, визначення конкурентоспроможних преимуществ.	Впровадження стратегій маркетингу, які підсилюють унікальність продукту, участь у програмах лояльності.
За галузевою ознакою: внутрішньогалузева	Різний рівень конкурентів в різних сегментах ринку або спеціалізація на конкретних секторах.	Спрямування зусиль на розвиток у специфічних галузях, співпраця з ключовими гравцями у визначених секторах.
Конкуренція за видами товарів: товарно-родова	Конкуренція за певними характеристиками товарів або альтернативами.	Диференціація продукту, пошук унікальних властивостей, підвищення якості для виділення серед конкурентів.

Після визначення конкурентного середовища та його особливостей, а також їх впливу на діяльність компанії при розробці стартапу, необхідно провести детальний аналіз умов конкуренції за М. Портером:

Таблиця 5.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	**Прямі конкуренти**	**Потенційні конкуренти**	**Постачальники**	**Клієнти**	**Товарозамінники**
	Snopes	Специфіка ринку	Методи системи та аналітики	Контроль якості	Якісні та дешеві
Висновки	Боротьби не передбачається	Є конкуренти	Не диктують	Не диктують	Не присутні

Маючи інформацію за конкурентами, було сформовано перелік факторів конкурентноспроможності (таблиця 5.10).

Таблиця 5.10 – Фактори конкурентноспроможності

Фактор конкурентноспроможності	**Обгрунтування**
Технологічна інноваційність	Здатність впроваджувати нові технології і інновації, що покращують продукт або послугу та дозволяють виходити вперед конкурентів.
Якість обслуговування клієнтів	Високий рівень обслуговування та задоволення клієнтів, що визначається ефективною комунікацією, швидким вирішенням проблем та індивідуальним підходом.
Маркетингова стратегія	Ефективна стратегія маркетингу, яка визначається правильною позиціонуванням продукту, вивченням ринкових тенденцій та вдалою комунікацією.

Порівнюючи обгрунтування факторів конкурентноспроможності (технологічна інноваційність, якість обслуговування клієнтів, маркетингова

стратегія), є змога сформувати аналіз факторів конкурентноспроможності. Деталі у таблиці 5.11.

Таблиця 5.11 – Порівняльний аналіз сильних та слабких сторін системи

Фактор конкурентноспроможності	**Бали 1-20**	**Рейтинг товарів конкурентів**						
		-3	-2	-1	0	1	2	3
Технологічна інноваційність							✓	
Якість обслуговування клієнтів						✓		
Маркетингова стратегія			-					

Маємо SWOT-аналіз продукту (таблиця 5.12):

Таблиця 5.12 – SWOT-аналіз продукту

Сильні сторони Технологічна інноваційність Широкий технічний стек	**Слабкі сторони** Обмежені ресурси
Можливості Ринковий ріст у сфері боротьби з дезінформацією	**Загрози** Обмежені ресурси для масштабування

SWOT-аналіз допомагає визначити всі сильні та слабкі сторони, можливості та загрози продукту, які пов'язані з плануванням стартап-продукту.

Наступним кроком є проектування альтернативної ринкової поведінки для інтеграції стартап-проекту на ринок та приблизний час реалізації комплексу (таблиця 5.13).

Таблиця 5.13 – Альтернативи ринкового впровадження стартап-проекту.

Альтернативи ринкового впровадження стартап проекту	**Ймовірність отримання ресурсів**	**Строки реалізації**
Швидкий вихід з MVP	50%	7 місяців
Поступовий вихід з продуктом	100%	2 роки

Отже, було проведено аналіз ринку та продукту. Відповідно до результатів проведеного конкурентного аналізу, факторів ринку та описання ідеї та характеристик стартап-проекту, можна зробити висновок, що існують досить сприятливі умови.

5.4 Розробка ринкової стратегії стартап-проекту

Для розробки ринкової стратегії продукту, необхідно проаналізувати цільову аудиторію (таблиця 5.14).

Таблиця 5.14 – Вибір цільових груп потенційних споживачів

Опис профілю цільової групи	**Цільова аудиторія**	**Опис профілю цільової групи потенційних клієнтів**	**Інтенсивність конкуренції у сегменті**	**Простота входу у сегмент**
Медійні компанії та видання	Висока	90%	Середня	Середня
Освітні установи та дослідницькі організації	Висока	10%	Низька	Низька

Обрано цільову групу №1. Визначимо базову стратегію розвитку продукту (таблиця 5.15).

Таблиця 5.15 – Базова стратегія розвитку

Обрана альтернатива розвитку проекту	**Стратегія охоплення ринку**	**Ключові конкурентоспроможні позиції відповідно до обраної альтернативи**	**Базова стратегія розвитку**
1	Концентрованого маркетингу	Простота системи	Стратегія спеціалізації

Щоб працювати у обраних сегментах, необхідно сформувати базову стратегію розвитку (таблиці 5.16 та 5.17).

Таблиця 5.16 – Базова стратегія конкурентної поведінки

Чи є проект "першопрохідцем" на ринку	**Чи буде компанія шукати нових споживачів, або забирати існуючих конкурентів**	**Чи буде компанія копіювати основні характеристики товару конкуренту, і які?**	**Стратегія конкурентної поведінки**
Ні	Так	Так	Наслідування лідеру

Таблиця 5.17 – Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	**Базова стратегія розвитку**	**Ключові конкурентноспроможні позиції власного стартап-проекту**	**Вибір асоціацій, які мають формувати комплексту позицію власного проекту**
Швидкість і простота	Концентрованого маркетингу	Проста та зручна у налаштуванні система	Якість, простота і спеціалізація

Визначивши стратегію позиціонування, маємо змогу розробити маркетингову програму стартап-проєкту, за яким є можливість підвищення показників продажу та зацікавленості зі сторони потенційних клієнтів застосунку.

5.5 Розроблення маркетингової програми стартап проекту

Після комплексного аналізу, необхідно повноцінно описати ключові переваги концепції постійного товару. Концепція постійного товару дозволяє підтримувати стабільність у виробництві та реалізації, забезпечуючи постійний запас товарів із заздалегідь визначеними характеристиками, (таблиця 5.18).

Таблиця 5.18 – Ключові переваги концепції постійного товару

Потреба	**Вигода яку пропонує товар**	**Ключові переваги перед конкурентами**
Ефективне виявлення дезінформації	Підвищення рівня достовірності інформації в мас-медіа та соціальних мережах.	- Використання передових технологій для точного виявлення дезінформації.
Оптимізація обробки текстів	Автоматизація та полегшення аналізу великого обсягу інформації.	- Використання технічної інноваційності для швидкої та ефективної обробки текстових даних.
Система постійного вдосконалення	Можливість адаптації до нових видів дезінформації та змін у мовному аналізі.	- Регулярні оновлення та вдосконалення алгоритмів виявлення дезінформації.

Визначимо основну систему збуту, в межах якої приймається рішення (таблиця 5.19).

Таблиця 5.19 – Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	**Функції збуту, які має виконувати постачальник товару**	**Глибина каналу збуту**	**Оптимальна система збуту**
Система надаватиметься за підпискою	Продаж	0(напряму)	Власна або через посередника (сайт)

В рамках завершення маркетингової стратегії, важливою складовою є розробка концепції маркетингових комунікацій, що включає в себе систематичний аналіз та стратегічне планування ефективних методів взаємодії з цільовою аудиторією для досягнення максимального впливу та підтримки маркетингових цілей. Концепція маркетингових комунікацій наведена у таблиці 5.20.

Таблиця 5.20 – Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	**Канали комунікації**	**Ключові позиції**	**Завдання рекламного повідомлення**	**Концепціія рекламного звернення**
Купівля через веб-сайт	Інтернет	Підтримка платформ	Показати переваги простоти у використанні та необхідність	Відео, повідомлення, сайт

Висновки до розділу 5

У розділі було висвітлено розробку стратегії введення на ринок та маркетингової стратегії для неї. Зазначено, що цей ринок є вузьким і специфічним, з невеликою кількістю конкурентів, але він не надто сприятливий через потребу у спеціальному обладнанні. Оскільки запропонована система є універсальною та доступною на різних платформах, стартап має можливість вийти на ринок і стати гравцем в ньому.

Додатково були розглянуті сильні та слабкі сторони проекту за допомогою SWOT-аналізу, проведений аналіз конкурентів і визначена цільова аудиторія. На основі отриманих даних був сформований концепт маркетингової стратегії, орієнтованої на цільові аудиторії.

ЗАГАЛЬНІ ВИСНОВКИ

1. На основі проведених досліджень обгрунтовано використання нейронних мереж SBERT та XLNet Large для векторизації та класифікації текстового повідомлення, а також методу RAKE для передобробки текстового повідомлення.
2. Представлено обрані методи векторизації та класифікації тексту у моделях штучного інтелекту та виокремлено гіперпараметри кожної моделі.
3. Створено тестову вибірку (407 достовірних дописів з The Guardian та The New York Times, 750000 дописів з соціальної мережі X) для обчислювальних експериментів з виявлення дезінформаціїї на моделях штучного інтелекту. На основі обраних методів та бібліотек FastAPI, Sentence Transformers та React розроблено програмне забезпечення виявлення дезінформації в текстах.
4. Проведено обчислювальні експерименти для комбінацій з таких моделей: RoBERTa-base, All-MiniLM-L6-v2, all-mpnet-base-v2, XLNet-Large-Cased, bert-base-uncased. За результатами обчислювальних експериментів оптимальною за точністю є архітектура, яка складається з моделей All-MiniLM-L6-v2 та XLNet-Large-Cased.
5. Розроблено стартап-стратегію для проєкту, визначено слабкі та сильні сторони проєкту, складено маркетинговий концепт ідеї.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Holovakin M., Shapovalova S. Detecting disinformation in english-language posts on social media. *Technical sciences Innovative scientific research: theory and practice: Proceedings of the X International Scientific and Practical Conference*. November 21-24, 2023. Stockholm, P. 425-427. DOI:10.46299/ISG.2023.2.10.
2. Akuma S., Lubem T., Adom I. T. Comparing Bag of Words and TF-IDF with different models for hate speech detection from live tweets. *International Journal of Information Technology*. 2022. DOI:10.1007/s41870-022-01096-4.
3. Williams H. E. Book review: Information retrieval: Algorithms and heuristics, by David A. Grossman and Ophir Frieder. *Journal of the American Society for Information Science*. 2000. Vol. 51, no. 11. P. 1063–1064. DOI:10.1002/1097-4571(2000)51:11%3C1063::aid-asi1010%3E3.0.co;2-5.
4. Uribe D. Word Embeddings for Constructive Comments Classification. *Journal of Physics: Conference Series*. 2021. Vol. 1824, no. 1. P. 012005. DOI: 10.1088/1742-6596/1824/1/012005.
5. Suen C. Y. n-Gram Statistics for Natural Language Understanding and Text Processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1979. PAMI-1, no. 2. P. 164–172. DOI:10.1109/tpami.1979.4766902.
6. Landauer T., Dooley S. Latent semantic analysis. *the Conference, Boulder, Colorado*, 7–11 January 2002. Morristown, NJ, USA, 2002. DOI:10.3115/1658616.1658815.
7. Santos B. N. D., Marcacini R. M., Rezende S. O. Multi-Domain Aspect Extraction Using Bidirectional Encoder Representations From Transformers. *IEEE Access*. 2021. Vol. 9. P. 91604–91613. DOI:10.1109/access.2021.3089099.
8. Utami P. D., Sari R. Filtering Hoax Menggunakan Naive Bayes Classifier. *MULTINETICS*. 2018. Vol. 4, no. 1. P. 57–61. DOI:.32722/multinetics.v4i1.1179 .
9. Support Vector Machines. *Machine Learning*. 2011. P. 135–148. DOI:10.1201/9781420067194-9.

10. Shaules J. Deep Learning. *Language, Culture, and the Embodied Mind*. Singapore, 2019. P. 57–69. URL: https://doi.org/10.1007/978-981-15-0587-4_5.
11. Lappalainen H., Miskin J. W. Ensemble Learning. *Advances in Independent Component Analysis*. London, 2000. P. 75–92. DOI:10.1007/978-1-4471-0443-8_5.
12. Rigatti S. J. Random Forest. *Journal of Insurance Medicine*. 2017. Vol. 47, no. 1. P. 31–39. DOI:10.17849/insm-47-01-31-39.1.
13. Boehmke B., Greenwell B. Gradient Boosting. *Hands-On Machine Learning with R*. 2019. P. 221–246. DOI:10.1201/9780367816377-12.
14. Tung A. K. H. Rule-Based Classification. *Encyclopedia of Database Systems*. New York, NY, 2016. P. 1–4. DOI:10.1007/978-1-4899-7993-3_559-2.
15. Kaggle: Your Machine Learning and Data Science Community. *Kaggle: Your Machine Learning and Data Science Community*. URL: https://www.kaggle.com/
16. Disinformation and Fake News. *Kaggle: Your Machine Learning and Data Science Community*. URL: https://www.kaggle.com/datasets/corrieaar/disinformation-articles (date of access: 11.12.2023).
17. Головна - EU vs Disinfo. *EU vs Disinfo*. URL: https://euvsdisinfo.eu/ua/ (дата звернення: 11.12.2023).
18. News, sport and opinion from the Guardian's Europe edition | The Guardian. *News, sport and opinion from the Guardian's Europe edition | The Guardian*. URL: https://www.theguardian.com/europe (date of access: 11.12.2023).
19. The New York Times International - Breaking News, US News, World News, Videos. *The New York Times - Breaking News, US News, World News and Videos*. URL: https://www.nytimes.com/international/ (date of access: 11.12.2023).
20. FactCheck.org. *FactCheck.org*. URL: https://www.factcheck.org/ (date of access: 11.12.2023).
21. Snopes.com. Snopes.com | *The definitive fact-checking site and reference source for urban legends, folklore, myths, rumors, and misinformation*. URL: https://www.snopes.com/ (date of access: 11.12.2023).

22. NewsGuard - Transparent Reliability Ratings for News and Information Sources. *NewsGuard*. URL: https://www.newsguardtech.com/ (date of access: 11.12.2023).
23. Zhao H., Bai C. S., Zhu S. Automatic Keyword Extraction Algorithm and Implementation. *Applied Mechanics and Materials*. 2010. Vol. 44-47. P. 4041–4049. URL: https://doi.org/10.4028/www.scientific.net/amm.44-47.4041
24. Reimers N., Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China. Stroudsburg, PA, USA, 2019. URL: https://doi.org/10.18653/v1/d19-1410
25. Attention Is All You Need In Speech Separation / C. Subakan et al. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, ON, Canada, 6–11 June 2021. 2021. URL: https://doi.org/10.1109/icassp39728.2021.9413901
26. XLNet Transfer Learning Model for Sentimental Analysis / C. R. Dhivyaa et al. *2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS)*, Coimbatore, India, 14–16 June 2023. URL: https://doi.org/10.1109/icscss57650.2023.10169445
27. TIOBE Index - TIOBE. *TIOBE*. URL: https://www.tiobe.com/tiobe-index/
28. Welcome to Python.org. *Python.org*. URL: https://www.python.org/ FastAPI. FastAPI. URL: https://fastapi.tiangolo.com/ (date of access: 11.12.2023).
29. NLTK :: Natural Language Toolkit. *NLTK :: Natural Language Toolkit*. URL: https://www.nltk.org/
30. SentenceTransformers Documentation – Sentence-Transformers documentation. *SentenceTransformers Documentation – Sentence-Transformers documentation.* URL: https://www.sbert.net/
31. Hugging Face – The AI community building the future. *Hugging Face – The AI community building the future*. URL: https://huggingface.co/

32. JavaScript | MDN. *MDN Web Docs*. URL: https://developer.mozilla.org/ru/docs/Web/JavaScript.
33. React. *React*. URL: https://react.dev/ (date of access: 11.12.2023).
34. Bootstrap. *Bootstrap · The most popular HTML, CSS, and JS library in the world.* URL: https://getbootstrap.com/
35. Storybook. *Storybook*. URL: https://reactstrap.github.io/?path=/story/home-installation--page.
36. Interactive charting library | Highcharts. *Highcharts Blog | Highcharts*. URL: https://www.highcharts.com/
37. react-wordcloud. *npm*. URL: https://www.npmjs.com/package/react-wordcloud.
38. Elasticsearch: The Official Distributed Search & Analytics Engine | Elastic. *Elastic*. URL: https://www.elastic.co/elasticsearch.
39. Method of Ontology Use in OODA / V. Lytvyn et al. 2020 *IEEE Third International Conference on Data Stream Mining & Processing (DSMP)*, Lviv, Ukraine, 21–25 August 2020. 2020. URL: https://doi.org/10.1109/dsmp47368.2020.9204107.

Додаток А

Визначення дезінформації в текстах засобів масової інформації та соціальних мереж

Апробація

The 10th International scientific and practical conference “Innovative scientific research: theory and practice”

Аркушів 5

International Science Group

ISG-KONF.COM

X

INTERNATIONAL SCIENTIFIC AND PRACTICAL CONFERENCE "INNOVATIVE SCIENTIFIC RESEARCH: THEORY AND PRACTICE"

Stockholm, Sweden

November 21-24, 2023

ISBN 979-8-89238-622-7

DOI 10.46299/ISG.2023.2.10

91.	Andrushchak I., Koshelyuk V., Kominko V., Shepeliuk S., Levchuk M. PROTECTION OF WEBSITE CONTENT TYPE OF UNAUTHORIZED ACCESS	409
92.	Doikov V. APPLICATION OF DEEP NEURAL NETWORKS IN THE DIAGNOSIS OF GLAUCOMA	414
93.	Geidarov P.S.O. GENERAL PRINCIPLES OF THE ALGORITHM FOR ANALYTICAL COMPUTATION OF WEIGHTS OF CONVOLUTIONAL NEURAL NETWORKS	420
94.	Holovakin M., Shapovalova S. DETECTING DISINFORMATION IN ENGLISH-LANGUAGE POSTS ON SOCIAL MEDIA	425
95.	Juergen K., Oliinyk Y., Oberemok Z. UKRAINE-GERMANY: COOPERATION IN THE FIELD OF ENERGY SAVING	428
96.	Kazimov F.S. UNCERTAINTY OF INTERLABORATORY MEASUREMENTS. INTRODUCTION OF UNCERTAINTY OF INTERLABORATORY MEASUREMENTS INTO INTERNAL CALCULATIONS OF THE LABORATORY TO INCREASE THE ACCURACY OF MEASUREMENTS IN CONDITIONS OF INSUFFICIENT INFORMATION	435
97.	Petrov K. MUSIC AND PROBABILITY	440
98.	Petrovskyi O. STYLE DESCRIPTORS AWARE FONT COMPLETION	445
99.	Zeleneva I., Holub T., Hrushko S., Shadrin M. PARALLEL ORGANISATION OF AN EMBEDDED SYSTEM CONTROL ON A SINGLE FPGA CHIP	453

DETECTING DISINFORMATION IN ENGLISH-LANGUAGE POSTS ON SOCIAL MEDIA

Holovakin Mykyta,
Student
Igor Sikorsky Kyiv Polytechnic Institute,
mholovakin@gmail.com

Shapovalova Svitlana,
Ph.D., Associate Professor
Igor Sikorsky Kyiv Polytechnic Institute,
lanashape@gmail.com

In the midst of military conflicts characterised by instability and threats to national and international security, the strategic use of disinformation as a powerful tool of influence has become increasingly common. As disinformation spreads through media and social networks during wartime, the need to identify and counter misleading narratives has grown significantly.

Advances in text vectorisation techniques have been instrumental in addressing this challenge. From statistical approaches such as Bag-of-Words and TF-IDF [1], the evolution has led to more sophisticated methods such as Word Embeddings (Word2Vec, GloVe), Doc2Vec, and transformative models such as BERT and GPT. These techniques serve as indispensable tools for deciphering the intricate web of disinformation that is prevalent in wartime communications.

Recent trends in the field emphasise the importance of contextualisation. Models are now being pre-trained on large corpora and fine-tuned for specific tasks, making them more attuned to the complexities of wartime language use. In addition, ongoing developments focus on improving interpretability and removing bias to ensure that these models remain adaptable and effective in the dynamic landscape of identifying and countering disinformation in times of conflict.

The dataset is constructed by collecting tweets that contain hashtags related to key events, such as #WarInUkraine. This targeted approach allows for a focus on discussions directly related to key issues, providing a nuanced perspective on the information landscape surrounding these events. Tweet data consists of usernames, user IDs, post IDs, URLs, content, quoted content, replies, hashtags, like counts and reply counts. In the pursuit for accurate and reliable information, authoritative sources are used such as “The Times” and “The Guardian” as benchmarks for truth. These reputable news outlets form the basis of our analysis, allows to assess the veracity of claims and identify potential misinformation within the dataset. To enhance contextual understanding within the analysis, sources of truth are embedded directly into Elasticsearch documents [2]. This integration ensures that the information retrieval process powered by Elasticsearch includes content from trusted sources, providing a

more complete view of the information landscape. Elasticsearch serves as a powerful tool for retrieving articles related to specific keywords extracted from tweets. Queries are built against the headline field of indexed documents, efficiently filtering out relevant information.

The overall process involves refining and enhancing textual data for improved analysis and modelling. The data undergoes steps to remove quotes and handle emoticons to ensure a cleaner representation. This is followed by tweet normalisation, which converts user mentions and URLs into standardised tokens, taking into account contractions and specific character replacements.

In this stage, SBERT [3], a pre-trained language model designed for sentence embedding, is used to capture the semantic meaning of tweets and articles. Text is transformed into vectors while preserving semantic relationships. The chosen SBERT model, "all-MiniLM-L6-v2" [4], operates in a vector space of dimension 384, allowing efficient representation of tweet and article content with a maximum token size of 256. To assess the relationship between tweets and associated articles, XLNet Large, a pre-trained language model fine-tuned for sequence classification tasks, is used. Pairs of sentences are categorised as either entailing, contradicting or having no clear relationship.

The confusion matrix is shown in Figure 1:

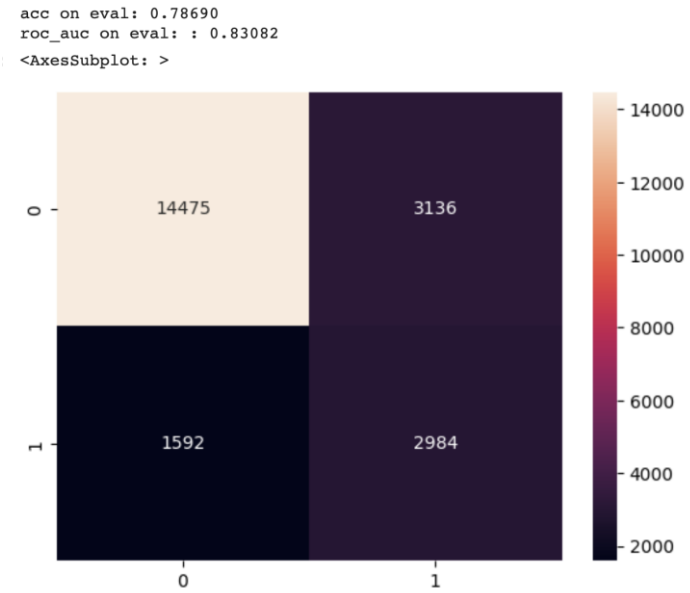

Figure 1. Confusion Matrix

The overall accuracy of the model on the training set is 0.78 or 78%. This metric represents the proportion of correctly classified instances out of the total.

The ROC AUC value is 0.83, indicating a good ability of the model to discriminate between positive and negative instances. Higher values indicate better discrimination between classes.

Conclusion

1. The study concludes that the method utilizing the "all-MiniLM-L6-v2" neural network model SBERT effectively detects misinformation in social media messages.

2. A dataset has been generated using the developed parser, containing information such as usernames, user IDs, post IDs, URLs, content, quoted content, replies, hashtags, like counts, and reply counts.

3. Based on the results of the computational experiments, the metric for accurate classification is 78%, indicating that 78% of instances were correctly classified on the training set. Additionally, the ROC AUC value is 0.83, signifying a good ability of the model to distinguish between positive and negative instances. A higher ROC AUC value suggests better discrimination between classes.

References

1. Akuma S., Tyosar L., Isaac T., Adom I. T. Comparing Bag of Words and TF-IDF with different models for hate speech detection from live tweets. *International Journal of Information Technology*. 2022. No: 14.7. P. 3629-3635. DOI:10.1007/s41870-022-01096-4.
2. Andhavarapu A. Learning Elasticsearch. Packt Publishing Ltd, 2017. 374 p.
3. Reimers N., Gurevych I. Sentence-bert: Sentence embeddings using siamese bert-networks. *9th International Joint Conference on Natural Language Processing*. 2019. P. 3973-3983. DOI:10.18653/v1/D19-1410.
4. Model card: sentence-transformers/all-MiniLM-L6-v2. *Hugging Face*. URL: https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2.