НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
Факультет прикладної математики
Кафедра прикладної математики

«До захисту допущено»
Завідувач кафедри
_____ Олег Чертов
«___» _____ 2023 р.

Дипломна робота

на здобуття ступеня бакалавра
за освітньо-професійною програмою «Наука про дані та математичне моделювання»
спеціальності 113 «Прикладна математика»
на тему: «Інтелектуальний аналіз тексту на вміст інформаційно-психологічних операцій за допомогою методів глибинного навчання»

Виконала:
студентка IV курсу, групи КМ-91
Неділько Дарина Вікторівна

_____

Керівник:
Доцент, канд. фіз.-мат. наук, доцент
Третиник Віолета Вікентіївна

_____ _____

Консультант з нормоконтролю:
Старший викладач,
Мальчиков Володимир Вікторович

_____

Рецензент:
Доцент каф. ММСА, канд. техн. наук, доцент
Дідковська Марина Віталіївна _____

Засвідчую, що в цій дипломній роботі немає запозичень із праць інших авторів без відповідних посилань.
Студентка _____

Київ — 2023 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет прикладної математики
Кафедра прикладної математики

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 113 «Прикладна математика»

Освітньо-професійна програма «Наука про дані та математичне моделювання»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олег Чертов

«___» _____ 2023 р.

ЗАВДАННЯ

на дипломну роботу студентці

Неділько Дарині Вікторівні

1. Тема роботи: «Інтелектуальний аналіз тексту на вміст інформаційно-психологічних операцій за допомогою методів глибинного навчання», керівник роботи Третиник Віолета Вікентіївна, канд. ф.-м. наук, доцент, затверджені наказом по університету від «31» травня 2023 р. № 2108-С.
2. Термін подання студенткою роботи: «12» червня 2023 р.
3. Вихідні дані до роботи: розроблювана система повинна аналізувати текст та визначати наявність вмісту інформаційно-психологічних операцій, мінімальна точність розпізнавання — 70%.
4. Зміст роботи: виконати аналіз існуючих методів розв’язання задачі, вибрати метод інтелектуального аналізу тексту, спроектувати автоматизовану систему розпізнавання ІПСО у текстовій інформації, здійснити програмну реалізацію розробленої системи, провести тестування розробленої системи.
5. Перелік ілюстративного матеріалу: архітектурні схеми нейроних мереж, блок-схеми розроблених алгоритмів, схема взаємодії модулів системи, знімки екранних форм.

6. Дата видачі завдання: «06» лютого 2023 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Вивчення та збір літератури за тематикою дипломної роботи та збір даних	13.11.2022	
2	Проведення порівняльного аналізу математичних методів інтелектуального аналізу тексту	11.12.2022	
3	Проведення порівняльного аналізу математичних методів інтелектуального аналізу тексту на наявність ІПСО, неправдивої інформації або провокацій	25.12.2022	
4	Підготовка матеріалів першого розділу роботи	01.02.2023	
5	Розроблення математичного забезпечення задачі інтелектуального аналізу тексту на вміст інформаційно-психологічних операцій	01.03.2023	
6	Підготовка матеріалів другого розділу роботи	19.03.2023	
7	Підготовка матеріалів третього розділу роботи	05.04.2023	
8	Розроблення програмного забезпечення задачі інтелектуального аналізу тексту на вміст інформаційно-психологічних операцій	15.04.2023	
9	Підготовка матеріалів четвертого розділу роботи	03.05.2023	
10	Оформлення пояснювальної записки	01.06.2023	

Студентка _____ Дарина НЕДІЛЬКО

Керівник роботи _____ Віолета ТРЕТИНИК

АНОТАЦІЯ

Дипломну роботу виконано на 89 аркушах, вона містить 2 додатки та перелік посилань на використані джерела з 25 найменування. У роботі наведено 34 рисунки та 6 таблиць.

Метою даної дипломної роботи є створення інструменту для інтелектуального аналізу тексту на вміст інформаційно-психологічних операцій за допомогою методів глибинного навчання.

У роботі проведено аналіз існуючих рішень указаної задачі за допомогою глибинного навчання методами навчання з учителем, без учителя, гібридними та з використанням підкріплення. Виконано їх порівняння з погляду точності отримуваних розв'язків та ефективності алгоритмів. Для розв'язання задачі в роботі вибрано гібридну модель BERT, а саме її модифікацію RoBERTa.

Для тренування моделі було підготовлено набір даних, що містить новини з різних джерел. Дані були розмічені відповідно до змісту новин. Розроблено автоматизовану систему, що реалізує обраний метод. Розроблено зручний інтерфейс. Виконано тестування розробленої системи.

Ключові слова: ІПСО, глибинне навчання, машинне навчання, нейронні мережі, перевірка інформації, RoBERTa, довга-короткочасна пам'ять, двонапрямкові трансформери, варіаційні автоенкодери, метод опорних векторів.

ABSTRACT

The thesis is presented in 89 pages. It contains 2 appendixes and bibliography of 27 references. Thirty-four figures and 6 tables are given in the thesis.

The goal of the thesis is to develop a tool for solving the problem of machine-based text analysis of psychological operations detection.

The thesis analyzes existing solutions to the specified task using deep learning methods such as supervised learning, unsupervised learning, hybrid approaches, and reinforcement learning. A comparison is made in terms of the accuracy of the obtained results and the efficiency of the algorithms. The hybrid model BERT, specifically its modification RoBERTa, is chosen for solving the task in this thesis.

A dataset containing news articles from various sources is prepared for training the model. The data is labeled according to the content of the news. An automated system is developed to implement the chosen method, along with a user-friendly interface. Testing of the developed system is performed.

Keywords: Psychological Operations (PSYOPs), deep learning, machine learning, neural networks, information verification, RoBERTa, long short-term memory (LSTM), bidirectional transformers, variational autoencoders, support vector machines (SVM).

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

(M)VAE - (Multimodal) Variational Autoencoder

BERT - Bidirectional Encoder Representations from Transformers

DL (ГН) – Deep Learning (Глибинне навчання)

LSTM – Long-Short Time Memory

NBC – Naïve Bayes Classifier

NLP (ОПМ) – Natural Language Processing (Обробка Природнього Мовлення)

NN (НМ) – Neural Networks (Нейроні Мережі)

PSYOPS (ІПСО) - Psychological operation (Інформаційно-Психологічна Операція)

RNN (РНН) – Recurrent Neural Networks (Рекурентні Нейроні Мережі)

ВСТУП

В сучасному світі, зокрема у сьогоднішній реальності українців, швидкий доступ до великої кількості інформації є надзвичайно важливим. Проте, не менш важливим є розуміння та оцінка цієї інформації. У цьому контексті інтелектуальний аналіз тексту стає актуальним. Однією з ключових складових такого аналізу є виявлення інформаційно-психологічних операцій (ІПСО), які можуть включати в себе маніпулювання, зміну уваги, формування стереотипів та інші прийоми впливу на емоційний стан людини.

Проте, необхідно зазначити, що ІПСО можуть мати негативний вплив на людей та суспільство. Такі операції зазвичай використовуються для досягнення певних цілей, що може бути шкідливим для індивідів та суспільства в цілому. Наприклад, маніпуляції з думками та поведінкою людей можуть спричиняти негативні наслідки, такі як спотворення реальності, підтримка дискримінації, відчуття страху або паніки, введення в оману та інше.

У зв'язку з цим, розробка математичного та програмного забезпечення, яке буде визначати наявність ІПСО у текстах та надавати корисну інформацію про це, може бути важливим кроком у забезпеченні інформаційної безпеки та етичності використання інформації. Такий інструмент зможе допомогти користувачам розрізняти маніпулятивну та неманіпулятивну інформацію, підвищувати їх критичність та свідомість та знижувати можливість впливу маніпулятивних ІПСО на їхні рішення та дії.

У цій роботі буде досліджено можливості методів глибинного навчання для розробки програмного забезпечення, яке зможе виявляти наявність ІПСО у текстах та допомагати в оцінці їх впливу. Дослідження має на меті розробити ефективний та корисний інструмент для інтелектуального аналізу тексту, що забезпечить захист від шкідливих впливів та сприятиме зростанню критичності та етичності використання інформації.

1 ПОСТАНОВКА ЗАДАЧІ

Метою даного дипломного проекту є розробка математичного та програмного забезпечення для інтелектуального аналізу тексту з метою виявлення наявності інформаційно-психологічних операцій (ІПСО) та оцінки їх впливу на читачів за допомогою методів глибинного навчання. Для досягнення цієї мети передбачається вирішення наступних задач:

1. Зібрати та підготувати корпус текстів, що містять ІПСО/знайти попередньо підготовлені корпуси.
2. Розробити та навчити модель глибинного навчання для виявлення ІПСО у текстах.
3. Реалізувати програмне забезпечення, що використовує навчену модель для виявлення ІПСО.
4. Провести тестування розробленого програмного забезпечення на різних даних та оцінити його ефективність.
5. Розробити інтерфейс для взаємодії з програмним забезпеченням та додати функції з покращення взаємодії та користувацького досвіду.

Основні вимоги до розробленої системи інтелектуального аналізу тексту на вміст інформаційно-психологічних операцій:

a) Швидкодія: Система має бути ефективною, здатною обробляти великі обсяги текстових даних в реальному часі.

b) Точність: Розроблена модель повинна виявляти інформаційно-психологічні операції з високою точністю.

c) Гнучкість: Система повинна бути гнучкою та адаптованою до різних типів текстів та розпізнавати різноманітні інформаційно-психологічні операції.

d) Масштабованість: Система повинна бути здатною працювати з великими обсягами даних та масштабувати ресурси для ефективного виконання завдань.

e) Інтерфейс та взаємодія: Користувачі повинні мати зручний та інтуїтивно зрозумілий інтерфейс системи, а також отримувати зрозумілу та зворотну інформацію про виявлені інформаційно-психологічні операції.

f) Безпека: Система повинна гарантувати конфіденційність та захист персональних даних користувачів.

2 ПРЕДМЕТ ДОСЛІДЖЕННЯ ТА АКТУАЛЬНІСТЬ

2.1 Опис предмету дослідження

Інформаційно-психологічні операції (ІПСО) є складним та різнобічним явищем, що використовується для впливу на емоційний стан, думки, переконання та поведінку людини. Це поняття виникає в контексті інформаційного суспільства, де доступ до інформації стає все ширшим та безпосереднім.

ІПСО складаються з різних компонентів, які використовуються для досягнення бажаного впливу. Одна з ключових складових ІПСО - маніпуляція, що передбачає використання психологічних методів та стратегій з метою зміни переконань, думок та емоцій людей. Маніпуляції можуть спрямовуватись на створення позитивного ставлення до конкретного продукту або ідеї, підтримку певних політичних переконань або спотворення фактів з метою досягнення конкретних цілей.

Іншою складовою ІПСО є зміна уваги, яка полягає в впливі на увагу людини з метою привернення до певного контенту або відволікання від іншого. Наприклад, використання яскравих кольорів, привабливих зображень або емоційно заряджених заголовків може привернути увагу та привести людину до сприйняття певного матеріалу.

Формування стереотипів також є складовою ІПСО. Це означає створення усталених, загальних уявлень про певні групи людей або ідеї, що можуть впливати на сприйняття та оцінку інформації. Стереотипи можуть мати негативний, дискримінаційний або спотворюючий характер, що може призводити до негативного впливу на сприйняття та поведінку людей.

Окрім того, фейкові новини або недостовірна інформація є також важливою складовою ІПСО. Це означає навмисне поширення неправдивої або маніпулятивної інформації з метою впливу на громадську думку, формування певних уявлень або спотворення фактів. Фейкові новини можуть використовуватись для створення

емоційних реакцій, паніки, підтримки певних політичних сил або навіть спотворення сприйняття подій.

Розуміння складових ІПСО та розробка ефективних методів їх виявлення та протидії є важливим завданням для забезпечення інформаційної безпеки та розвитку критичного мислення у суспільстві.

2.2 Методологія розпізнавання ІПСО

Розпізнавання ІПСО (інформаційно-психологічних операцій) людиною може вимагати деякої кваліфікації та уважності, оскільки ІПСО має тенденцію бути хитроманіпулятивним і прихованим. Щоб ефективно розпізнавати ІПСО, слід приділити увагу кільком аспектам.

1. Критичне мислення: Свідоме споживання інформації та фільтрування через постановку питання щодо достовірності та об'єктивності висловлювань. Критичне мислення включає здатність аналізувати аргументи, виявляти логічні неузгодженості та шукати об'єктивні докази підтвердження. Це допомагає пильно спостерігати за інформацією, що надходить, і розуміти, коли можуть бути застосовані маніпулятивні техніки.

2. Уважне читання та слухання: ІПСО часто намагається вплинути на емоційний стан та переконання людей. Уважне читання і слухання допомагають виявити незвичайні або надмірно емоційні висловлювання, які можуть бути спрямовані на маніпуляцію. Звертання уваги на загальний контекст і аргументацію допомагає оцінити, наскільки обґрунтовані та зрозумілі висловлювання.

3. Перевірка джерел інформації: Важливо перевіряти надійність джерел інформації, особливо в сучасному цифровому середовищі, де новини швидко поширюються. Переконайтеся, що джерело має достатню авторитетність і

репутацію. Додатковою перевіркою може бути пошук інформації з кількох джерел, щоб переконатися, що інформація є консистентною та надійною.

4. Розуміння маніпуляційних технік: ІПСО часто використовує різні маніпуляційні техніки, такі як перекручення фактів, використання емоційного навантаження, створення фальшивих авторитетів тощо. Знання цих технік допомагає розпізнавати ознаки маніпуляції та реагувати на них адекватно.

Ці підходи разом розвивають навички розпізнавання ІПСО і допомагають бути більш обережним та критичним споживачем інформації. Важливо постійно вдосконалювати ці навички, оскільки ІПСО може змінюватися та адаптуватися до нових ситуацій.

2.3 Інструменти вирішення задачі розпізнавання ІПСО

Для розпізнавання ІПСО (інформаційно-психологічних операцій) можна використовувати різні інструменти та підходи. До них належать:

1. Математичне моделювання та аналіз текстів: Застосування методів обробки природньої мови та статистичного аналізу текстів може допомогти виявити ключові ознаки, закономірності та стилістику, пов'язані з ІПСО. Це може включати аналіз семантики, синтаксису, вживання емоційно заряджених слів та інші прийоми.

2. Машинне навчання: Використання алгоритмів машинного навчання дозволяє створити моделі, які можуть розпізнавати ІПСО на основі великої кількості даних. Моделі можуть бути навчені розпізнавати маніпулятивні техніки, фейкові новини, стилістичні особливості та інші ознаки ІПСО. Моделі можуть бути натреновані як на підставі попередньо розмічених даних, так і на основі постійної взаємодії з користувачами та їхнього відгуку.

3. Соціальна аналітика: Аналіз активності у соціальних мережах та медіа може допомогти виявити шаблони та тренди, пов'язані з ІПСО. Використання

алгоритмів машинного навчання та обробки даних дозволяє виявити впливових акторів, поширення фейкових новин, маніпулятивні кампанії та інші прояви ІПСО у соціальних медіа.

4. Експертний аналіз: Врахування думки та експертизи відповідних фахівців, які мають досвід у розпізнаванні ІПСО, може бути корисним. Експерти можуть допомогти виявити певні ознаки та шаблони, які не завжди легко виявити за допомогою автоматичних інструментів.

2.4 Переваги обраного методу

Порівнюючи методи розпізнавання ІПСО, можна зробити висновок, що машинне навчання є одним з найефективніших методів, оскільки воно надає кращі результати та має деякі переваги порівняно з іншими підходами. З найбільш вагомими є наступні:

1. Висока точність: Машинне навчання дозволяє створити моделі, які можуть навчатися на великій кількості даних та здатні досягати високої точності у розпізнаванні ІПСО. За допомогою алгоритмів машинного навчання моделі можуть виявляти навіть складні шаблони та зв’язки, що забезпечує кращу ефективність порівняно з іншими методами.

2. Автоматизація: Машинне навчання дозволяє автоматизувати процес розпізнавання ІПСО. Після того як модель навчена на відповідних даних, вона може працювати автономно та швидко аналізувати текстові дані на наявність ознак ІПСО. Це дозволяє економити час та ресурси, необхідні для постійного залучення великої кількості осіб.

3. Масштабованість: Машинне навчання може бути легко масштабоване для роботи з великим обсягом даних та швидко забезпечувати результати. Моделі машинного навчання можуть бути використані для обробки великої кількості

текстових даних з різних джерел, що робить їх ефективними в аналізі ІПСО на широкому масштабі.

Незважаючи на те, що машинне навчання може бути найкращим методом у багатьох випадках, варто пам'ятати, що комбінація різних методів зазвичай призводить до кращих результатів. У деяких випадках точний результат може вимагати комбінації машинного навчання з експертним аналізом або соціальною аналітикою.

2.5 Висновки до розділу

Інформаційно-психологічні операції стають все більш актуальною темою в сучасному світі, де інформаційне середовище широко поширене. Вплив нашого емоційного стану, переконань та поведінки через різноманітні інформаційні канали стає все більш інтенсивним. У такому контексті, виявлення та детекція ІПСО має велике значення для забезпечення інформаційної безпеки, захисту індивідуальних прав та розвитку критичного мислення.

Методи машинного навчання виявляються потужним інструментом для виявлення та аналізу ІПСО. З використанням глибинного навчання та нейронних мереж, ці методи дозволяють автоматично аналізувати текстовий контент, виявляти маніпулятивні стратегії та психологічні впливи. Це створює можливість швидкого виявлення та реагування на потенційно шкідливий вплив інформації.

Застосування методів машинного навчання в детекції ІПСО дозволяє розробляти ефективні алгоритми, які здатні автоматично розпізнавати підозрілі та маніпулятивні патерни в тексті. Це сприяє не лише захисту користувачів від потенційно шкідливих впливів, але і розвитку критичного мислення в суспільстві. Активне використання таких методів сприяє підвищенню інформаційної

грамотності, забезпеченню об'єктивності та достовірності інформації, що сприяє формуванню більш обізнаних та самостійних індивідів.

3 РОЗГЛЯД ГОТОВИХ РІШЕНЬ

3.1 Методи навчання з вчителем (Supervised learning)

Навчання з вчителем або ж supervised learning це навчання моделі на вже розмічених даних, тобто модель навчається на даних, де кожен набір або ж вектор має свою категорію. У зарубіжній літературі набір характеристик називають features, а визначеною для них категорією є labels або targets.

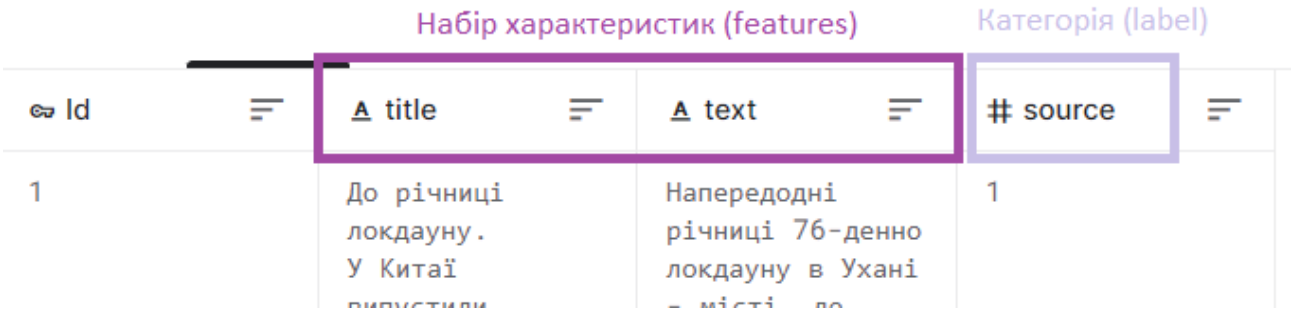


Рисунок 3.1.1 – Приклад розміченого датасету [1].

3.1.1 Рекурентні нейроні мережі (RNN)

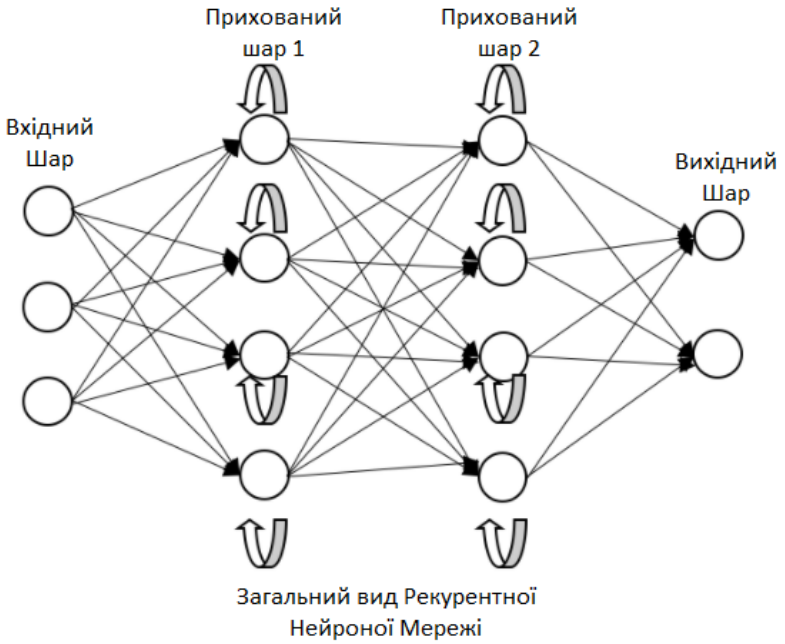
Рекурентні нейронні мережі (RNN) широко використовуються для розпізнавання Інформаційно-Психологічних Операцій (ІПСО). RNN є типом нейронної мережі, яка може аналізувати та розуміти послідовні дані, такі як текстові повідомлення.

У контексті розпізнавання ІПСО, RNN може бути використаний для аналізу та класифікації текстових даних з метою виявлення наявності ІПСО. Основна перевага RNN полягає в їх здатності враховувати контекстуальні залежності між словами та фразами у тексті [8].

Основний принцип роботи RNN полягає в тому, що вона зберігає попередні стани та використовує їх для обробки нових вхідних даних у послідовний спосіб. Вона проходить крізь кожен елемент послідовності та зберігає інформацію про попередні стани. Це дозволяє RNN засвоювати залежності та контекст у текстових даних [8].

Під час розпізнавання ІПСО, RNN може бути навчена на наборі даних з позначками, де кожен текстовий зразок має мітку, що вказує на наявність або відсутність ІПСО. Після навчання RNN може класифікувати нові текстові дані, проходячи їх через мережу та генеруючи вихід, який вказує на ймовірність наявності ІПСО. Застосовується поріг для прийняття рішення щодо класифікації тексту [6,7].

RNN мають свої обмеження, зокрема проблему зниклих градієнтів, яка може впливати на їх здатність моделювати довготривалі залежності. Поступова втрата попередніх станів під час обробки послідовних даних може бути значною проблемою [8]. Однак, існують вдосконалені архітектури, такі як LSTM (Long Short-Term Memory) і GRU (Gated Recurrent Unit), які вирішують цю проблему та поліпшують продуктивність RNN в аналізі тексту та розпізнаванні ІПСО.



Загальний вид Рекурентної
Нейроної Мережі

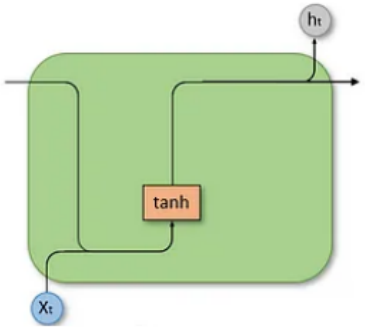


Рисунок 3.1.1.1 – Рекурентна Нейрона Мережа

3.1.2 Мережа довгої-короткочасної пам’яті (Long-Short Time Memory, LSTM)

LSTM (Довга короткочасна пам'ять) - це тип рекурентної нейронної мережі (RNN), яка часто використовується для моделювання послідовних даних, включаючи обробку природної мови. LSTM мережі добре підходять для виявлення ІПСО, оскільки вони ефективно виявляють залежності та шаблони в послідовних даних [7].

У виявленні ІПСО LSTM може використовуватися для аналізу та класифікації текстових даних з метою виявлення випадків ІПСО. Основною перевагою LSTM є його здатність моделювати довготривалі залежності в текстових даних, що є важливим для розуміння контексту та виявлення важливих шаблонів, пов'язаних з ІПСО [7].

Для роботи з LSTM попередньо необхідно підготувати дані. Вхідні текстові дані підлягають попередній обробці шляхом токенізації на окремі слова або підодиниці та кодування їх у числові представлення. Крім того, текст може пройти додаткові етапи попередньої обробки, такі як перетворення до нижнього регістру, видалення зайвих слів та стемінг/лематизація [7].

Надалі необхідно провести Word Embedding або ж Укладання Слів, у ході якого кожне слово в тексті представляється у вигляді щільного вектора. Укладання слів відображають семантичне значення та контекст слова, дозволяючи LSTM вивчати змістовні представлення.

LSTM модель складається з кількох шарів LSTM, які обробляють вхідну послідовність укладань слів. LSTM шари мають внутрішні клітини пам'яті, які зберігають і відновлюють інформацію протягом довгих послідовностей, що дозволяє їм фіксувати контекст та залежності [6].

Для навчання LSTM необхідні розмічені дані, де кожен текстовий зразок пов'язаний з бінарною міткою, що вказує на наявність або відсутність ІПСО. Модель

оптимізується за допомогою відповідної функції втрат, такою може бути бінарна кросс-ентропія(бінарна, оскільки у нас бінарний вихід з моделі), і оновлюється за допомогою зворотного поширення та градієнтного спуску [6].

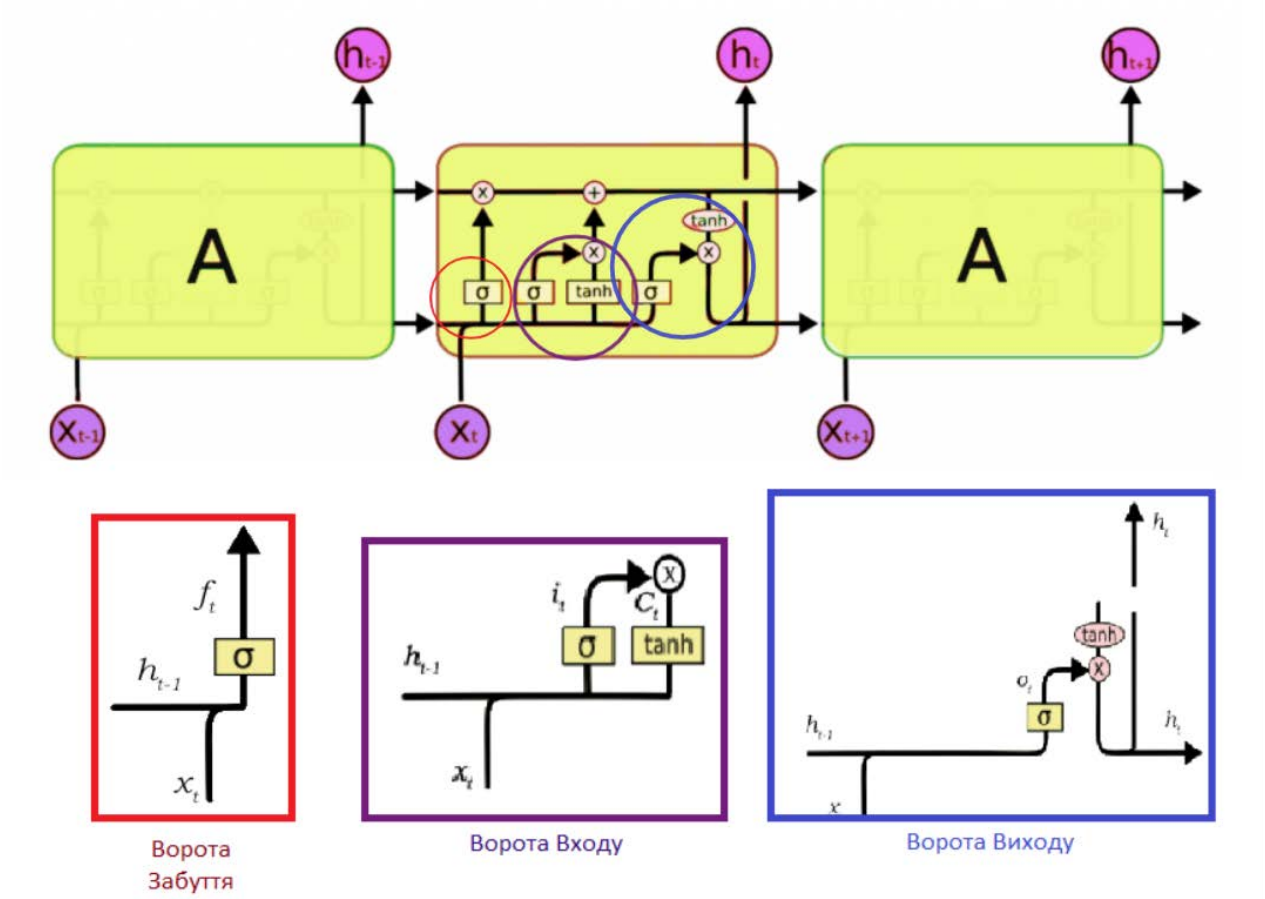


Рисунок 3.1.2.1 – Архітектура LSTM [5]

Після навчання LSTM модель може здійснювати прогнози на нових, невидимих текстових даних. Вона обробляє вхідну послідовність через свої LSTM шари і генерує прогноз, який вказує на ймовірність наявності ІПСО в тексті. Застосовується поріг для класифікації тексту як ІПСО чи не-ІПСО на основі передбаченої ймовірності.

LSTM мережі продемонстрували високу продуктивність в різних завданнях обробки природної мови, включаючи аналіз настрою, класифікацію тексту та генерацію мови. Їх здатність виявляти довготривалі залежності та моделювати

послідовні дані робить їх потужним інструментом для виявлення ІПСО в текстових даних.

3.1.3 Метод опорних векторів (Support Vector Machine)

Метод опорних векторів (SVM) є потужним алгоритмом для виявлення інформаційно-психологічних операцій (ІПСО). Він базується на навчанні з учителем і використовується для класифікації даних, де набори даних мають відповідні мітки [3]. Застосування SVM для виявлення ІПСО включає такі кроки:

1. Підготовка даних: Спочатку потрібно підготувати навчальні дані, які складаються з текстових документів або повідомлень, що потребують класифікації стосовно наявності ІПСО. Кожен документ повинен мати відповідну мітку, яка вказує, чи містить документ ІПСО або ні [4].
2. Векторизація даних: Текстові дані потрібно перетворити на числові вектори, щоб можна було застосувати їх у SVM. Це можна зробити шляхом використання методів, таких як TF-IDF або Bag-of-Words, які призначають числові значення словам або термінам у тексті [4].
3. Навчання SVM: Після підготовки даних можна навчити SVM за допомогою навчальних прикладів. SVM навчається визначати границю прийняття рішень, яка розділяє два класи - ІПСО та не-ІПСО. Метою SVM є знаходження гіперплощини в просторі ознак, яка максимізує відстань між двома класами [4].
4. Класифікація нових даних: Після тренування SVM можна застосовувати його для класифікації нових текстових документів або повідомлень. SVM використовує векторне представлення цих даних та раніше визначену границю прийняття рішень, щоб визначити, чи належить документ до класу ІПСО чи ні.

У роботі з цим методом варто зважати, що ефективність SVM у виявленні ІПСО залежить від якості підготовлених даних, вибору методів векторизації та налаштування параметрів SVM [4].

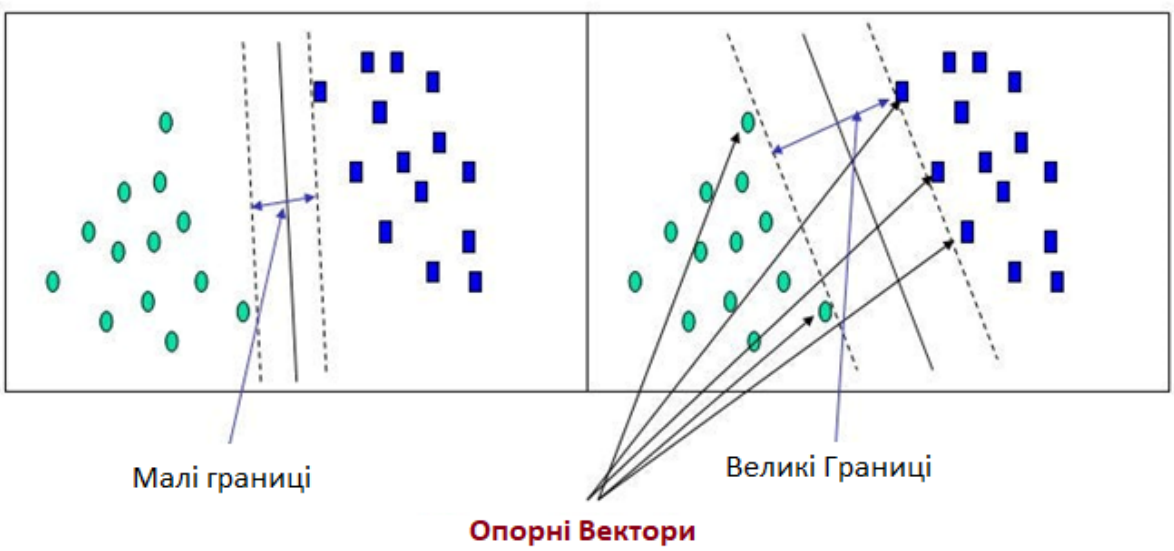


Рисунок 3.1.3.1 – Приклад границь у роботі Опорних Векторів

3.1.4 Наївний Баєсівський Класифікатор (Naïve Bayes classifier)

Наївний баєсівський класифікатор - це статистичний алгоритм, який використовує теорему Баєса для класифікації об'єктів на основі їх ознак. Математично ця теорема представляється у вигляді наступного рівняння [2]:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \qquad (3.1.4.1)$$

В контексті розпізнавання інформаційно-психологічних операцій (ІПСО), наївний баєсівський класифікатор може бути застосований для визначення, чи містить текстовий документ чи повідомлення ІПСО.

Основна ідея наївного баєсівського класифікатора полягає у використанні ймовірності для прийняття рішення про класифікацію. Він базується на наївному

припущенні про незалежність між ознаками, що означає, що кожна ознака вносить вклад незалежно в ймовірність належності до певного класу.

Таблиця 3.1.4.1 – Переваги та недоліки Наївного Баєсівського класифікатора для роботи з текстом

Переваги	Недоліки
Простота: NBC є простим і легко розумілим алгоритмом класифікації. Швидкодія: NBC має невелику обчислювальну складність і може швидко працювати з великими обсягами даних. Відсутність потреби в великій кількості тренувальних даних: NBC може працювати ефективно з обмеженою кількістю тренувальних даних.	Наївність припущення: NBC припускає незалежність ознак, що може бути недостатньо точним для складних текстових даних. Втрата контексту: NBC не враховує контекстуальні залежності між ознаками або словами в текстових даних. Вразливість до недостовірних ознак: NBC може неправильно працювати, якщо даний набір даних містить недостовірні або неадекватні ознаки. Обмежена здатність моделювання: NBC не здатний моделювати складні залежності або враховувати додаткові інформаційні шари в даних.

Для розпізнавання ІПСО наївний баєсівський класифікатор використовує текстові ознаки документа, такі як слова, фрази або інші лінгвістичні елементи [3]. Класифікатор будує модель, розраховуючи ймовірності належності документа до

кожного класу. Це вимагає обчислення двох компонентів: апріорної ймовірності класу і умовних ймовірностей ознак для кожного класу. Апріорна ймовірність класу - це ймовірність того, що документ належить до певного класу, заздалегідь, перед тим, як ми отримаємо будь-яку специфічну інформацію про сам документ. Ця ймовірність враховує загальний розподіл класів у популяції документів. Умовні ймовірності ознак вказують на ймовірність того, що певна ознака буде спостережена у документі при умові, що документ належить до певного класу. Ці ймовірності враховують статистичні зв'язки між класами та ознаками, що виявлені під час тренування моделі. Вони допомагають визначити, наскільки ймовірно певна ознака буде присутня у документі з конкретним класом.

Після будування моделі наївний баєсівський класифікатор може бути застосований до нових текстових документів для класифікації. Він обчислює ймовірності належності документа до кожного класу, використовуючи розраховані апріорні ймовірності та умовні ймовірності ознак. Документ класифікується до класу з найвищою ймовірністю.

Наївний баєсівський класифікатор (NBC) має свої плюси і мінуси при використанні для розпізнавання Інформаційно-Психологічних Операцій. Таблиця 3.1.4.1 перелічує особливості, на які необхідно зважати у разі роботи з цією моделю.

Отже, хоча NBC може бути корисним інструментом для розпізнавання ІПСО через його простоту та швидкодію, варто враховувати його обмежену здатність моделювання та вразливість до недостовірних ознак при використанні його в складних текстових даних.

3.2 Гібридні методи, методи навчання без вчителя

Гібридні методи - це машинного навчання поєднують два або більше різних підходів, моделей або алгоритмів машинного навчання з метою отримання кращих

результатів або вирішення більш складних завдань. Ці методи комбінують сильні сторони різних підходів і використовують їх у поєднанні, щоб досягти більшої точності, ефективності або розширити можливості моделі.

Одним з прикладів гібридного методу є комбінація моделей ансамблювання, таких як випадковий ліс або градієнтний бустинг, з нейронними мережами. В цьому випадку, нейронна мережа може використовуватись для витягування високорівневих ознак з даних, а потім ансамбль моделей може використовуватись для комбінування та прийняття рішень на основі цих ознак.

Зразком комбінації нейронних мереж і дерев рішень може слугувати метод, відомий як Neural Decision Forests (NDF). NDF поєднує сильні сторони нейронних мереж і дерев рішень для досягнення кращої точності та інтерпретованості моделі [9].

Інший приклад гібридного підходу - це комбінація методів безвчительового навчання (unsupervised learning) та навчання з вчителем (supervised learning). Безвчительове навчання може використовуватись для автоматичного витягування ознак або побудови репрезентації даних без використання міток, тоді як навчання з вчителем використовує мітки для навчання моделі класифікації або регресії. Комбінуючи ці два підходи, можна створити потужну модель, яка використовує навчання без вчителя для попереднього навчання або ініціалізації моделі, а потім навчання з учителем для профільного тренування моделі з використанням міток.

Гібридні методи машинного навчання надають можливість комбінувати різні підходи та моделі в залежності від потреб конкретної задачі. Це дозволяє поліпшити якість прогнозування, збільшити швидкодію моделі, розширити її можливості та отримати більш гнучкий підхід до вирішення завдань машинного навчання. За допомогою гібридних методів можна поєднувати сильні сторони різних моделей та підходів, використовуючи їх відповідно до потреб і характеристик конкретної задачі. Гібридні методи дозволяють підходити до складних задач з різноманітними типами даних та вимогами, надаючи більш гнучкі та ефективні рішення.

3.2.1 BERT – Модель представлення кодувальника з двонапрямковими трансформерами

BERT (Bidirectional Encoder Representations from Transformers) є потужною моделлю для розпізнавання інформаційно-психологічних операцій. ІПСО відносяться до стратегій маніпуляції інформацією з метою впливу на мислення, переконання та поведінку людей. Використання BERT у розпізнаванні ІПСО виявляється досить ефективним [10].

Архітектура BERT, заснована на трансформерах, дозволяє моделі аналізувати текстові дані з урахуванням контексту та залежностей між словами. BERT використовує механізм самоуваги для визначення важливих слів та взаємозв'язків між ними у тексті. Він також використовує механізм маскування для роботи з неповними даними та враховування контекстуальних залежностей [10].

Однією з головних переваг використання BERT у розпізнаванні ІПСО є його здатність розуміти семантику та синтаксис тексту, що дозволяє виявляти хитрі маніпуляції та зловживання у мові. Модель може розрізняти інформаційні заяви, що мають наміровано спотворений контекст або використовують підступні маніпуляції, від правдивих тверджень [11].

Також вагомою переваг BERT є те, що вона використовує контекстне уявлення слів. Вона здатна розуміти слова в залежності від контексту, в якому вони зустрічаються. Наприклад, слово "банк" може мати різні значення в залежності від контексту (банківська установа або берег річки). BERT може враховувати цей контекст і використовувати його для кращого розуміння семантики речень [13].

Проте, варто зазначити, що використання BERT також має деякі обмеження. По-перше, для тренування BERT потрібні значні обчислювальні ресурси та тривалий час. Це може становити проблему в деяких випадках, особливо якщо доступ до потужних обчислювальних систем обмежений [11].

Крім того, хоча BERT є потужною моделлю, він не є універсальним рішенням для всіх видів ІПСО. Деякі маніпуляції та спотворення можуть бути досить хитрими та складними для виявлення навіть для такої продуктивної моделі. Потрібно поєднувати BERT з іншими методами та моделями для забезпечення більшої ефективності в розпізнаванні ІПСО.

Загалом, BERT є потужним інструментом у боротьбі з ІПСО, завдяки його здатності аналізувати текст з урахуванням контексту та семантики. Його використання допомагає виявляти та розпізнавати маніпулятивні техніки у тексті. Однак, для досягнення кращих результатів, варто розглядати поєднання BERT з іншими моделями та методами розпізнавання ІПСО.

3.2.2 Варіаційні Автоенкодери (Vatiational Autoencoders, VAEs)

У статті про застосування мультимодальних варіаційних автоенкодерів [12] для розпізнавання фальшивих новин було розглянуто можливості одноіменної архітектури для роботи з текстовими даними у вигляді новин.

Мультимодальні варіаційні автоенкодери (Multimodal Variational Autoencoders) - це клас моделей штучного інтелекту, які використовуються для розпізнавання та аналізу Інформаційно-психологічних операцій (ІПСО). Ці автоенкодери поєднують у собі техніки мультимодального навчання та варіаційного автоенкодування для роботи з даними, які мають кілька модальностей (наприклад, текст, зображення, звук).

Головна мета мультимодальних варіаційних автоенкодерів - здатність моделювати складні залежності між різними модальностями даних та використовувати ці залежності для розпізнавання ІПСО. Вони навчаються відтворювати вхідні дані, представлені у вигляді кодів, що складаються з двох

частин: кодера, який перетворює вхідні дані у латентний простір, і декодера, який відтворює дані з латентного простору.

Застосування мультимодальних варіаційних автоенкодерів для розпізнавання ІПСО може включати аналіз даних різних модальностей, таких як тексти, зображення, аудіо, з метою виявлення характерних ознак та паттернів, пов'язаних з ІПСО. Це може допомогти як цивільним для розуміння чи варто споживати далі контент з ресурсу, так і військовим, щоб проаналізувати способи інформаційної боротьби ворога.

Мультимодальні варіаційні автоенкодери для розпізнавання ІПСО мають свої переваги та недоліки.

Застосування мультимодальних варіаційних автоенкодерів для розпізнавання ІПСО може сприяти поліпшенню розуміння взаємодії між різними модальностями даних та допомогти виявляти приховані залежності, що можуть бути корисні для прогнозування та запобігання ІПСО.

Таблиця 3.2.2.1 - Переваги та недоліки використання варіаційних автоенкодерів для розпізнавання тексту

Переваги	Недоліки
Гнучкість у використанні: Мультимодальні варіаційні автоенкодери можуть бути застосовані до різних сценаріїв розпізнавання ІПСО, включаючи військовий аналіз, розвідку, кримінальну діяльність тощо.	Потреба в великій кількості даних: Щоб мультимодальні варіаційні автоенкодери працювали ефективно, необхідно мати достатню кількість даних різних модальностей, що може бути важко зібрати або обробити.

Продовження таблиці 3.2.2.1

Переваги	Недоліки
Моделювання складних залежностей: Мультимодальні варіаційні автоенкодери дозволяють моделювати складні залежності між різними модальностями даних, що дозволяє розпізнавати та аналізувати ІПСО в більш точний спосіб. Урахування різноманітності даних: Ці автоенкодери дозволяють працювати з даними різних модальностей, таких як тексти, зображення, аудіо, що дозволяє врахувати різноманітні аспекти ІПСО та отримати більш повне розуміння ситуації.	Складність у навчанні та налаштуванні: Мультимодальні варіаційні автоенкодери є складними моделями, які вимагають ретельного навчання та налаштування. Це може вимагати значних обчислювальних ресурсів та експертного знання для досягнення оптимальних результатів. Втрата інтерпретованості: Використання глибоких нейромереж для розпізнавання ІПСО може призвести до втрати інтерпретованості результатів. Це може зробити складним зрозуміти, як саме модель прийшла до своїх висновків, що може бути проблемою з точки зору етики та права.

3.3 Методи навчання з підкріпленням (Reinforcement Learning)

Навчання з підкріпленням (reinforcement learning) є одним з підходів до машинного навчання, який використовується для навчання агента, який взаємодіє з оточенням і приймає рішення з метою максимізації певного числового показника продуктивності, який називається нагородою (reward). У цьому контексті агент

може бути комп'ютерною програмою, роботом або будь-яким іншим системою, яка вчиться взаємодіяти з навколишнім середовищем.

Основна ідея навчання з підкріпленням полягає в тому, що агент здійснює послідовні дії в оточенні, отримує нагороду або штрафи залежно від результатів своїх дій і навчається змінювати свою стратегію, щоб максимізувати загальну нагороду, яку він отримує протягом тренування.

Навчання з підкріпленням використовується в багатьох сферах, включаючи робототехніку, автономне водіння, гральну теорію та інші. У контексті розпізнавання ІПСО, навчання з підкріпленням може використовуватись для тренування моделей, які навчаються відрізняти фейкові новини від реальних, основуючись на нагородах, отриманих залежно від правильності їх класифікації.

Навчання з підкріпленням для роботи з новинами є не дуже поширеним рішенням, зокрема це пов’язано з тим, що для якісного тренування моделі необхідно мати живого спостерігача, що буде валідувати результат, що видає модель [12].

За допомогою навчання з підкріпленням можна розпізнавати Інформаційні Психологічні Впливи (ІПСО). Цей підхід має свої плюси та мінуси, і його використання залежить від конкретної ситуації та задачі.

Основні переваги навчання з підкріпленням:

— Адаптивність: Метод навчання з підкріпленням дозволяє агенту адаптуватися до змінюючого середовища та реагувати на нові типи ІПСО. Агент постійно взаємодіє з навколишнім середовищем, отримуючи з нього зворотний зв'язок, що дозволяє йому вдосконалювати свої стратегії та підходи до розпізнавання ІПСО.

— Гнучкість: Навчання з підкріпленням може пристосовуватися до різноманітних типів ІПСО. Агент може вчитися виявляти нові патерни, які вказують на ІПСО, навіть якщо вони раніше не були відомі. Це дає можливість ефективно боротися з постійно змінюючимися методами впливу та адаптуватися до нових загроз

— Оптимальне прийняття рішень: Навчання з підкріпленням дозволяє агенту знаходити оптимальні стратегії для розпізнавання ІПСО в реальному часі. Агент навчається максимізувати свій кумулятивний прибуток або мінімізувати збитки, що дозволяє зробити найефективніші рішення у відповідь на ІПСО.

Незважаючи на ці переваги, навчання з підкріпленням також має свої недоліки, через які в деяких випадках можуть бути використані інші методи. Ось кілька недоліків навчання з підкріпленням:

— Час і обчислювальні витрати: Навчання з підкріпленням може бути вимогливими щодо часу та обчислювальних ресурсів. Агент повинен проводити багато взаємодій з середовищем, щоб навчитися ефективним стратегіям. Це може бути обтяжливим процесом, особливо якщо існує велика кількість можливих ІПСО.

— Потреба в навчальних даних: Навчання з підкріпленням вимагає наявності відповідного середовища, у якому агент може навчатися. Це означає, що для розпізнавання ІПСО потрібні якісні навчальні дані, які відображають реальні ситуації та впливи.

— Ризик некоректного навчання: Якщо навчання з підкріпленням здійснюється неправильно або некоректно, це може призвести до незадовільних результатів. Недостатня кількість взаємодій з середовищем або некоректний зворотний зв'язок можуть призвести до недостатньої ефективності агента в розпізнаванні ІПСО.

У кінцевому підсумку, вибір методу для розпізнавання ІПСО залежить від специфічних обставин і вимог задачі. Навчання з підкріпленням може бути потужним і адаптивним підходом, який дозволяє ефективно боротися з різноманітними типами ІПСО, але вимагає витрат часу та ресурсів. Інші методи, такі як машинне навчання, можуть бути використані, коли навчальні дані доступні та методи більш підходять для конкретної задачі.

3.4 Порівняння методів

Комбіновані методи машинного навчання часто виявляються більш ефективними у розпізнаванні ІПСО порівняно з використанням окремих методів. Це пояснюється декількома факторами, які сприяють покращенню якості і точності розпізнавання. Комбінація різних методів машинного навчання дозволяє поєднати їхні переваги і уникнути їхніх недоліків, створюючи комплексний підхід, який покращує загальну ефективність системи розпізнавання. Кожен метод може мати свої сильні і слабкі сторони, але комбінування їх дозволяє отримати більше різноманітних рішень, які виокремлюють різні аспекти шахрайства і сприяють забезпеченню більш точного розпізнавання ІПСО. З комбінованих методів найбільш ефективним та простим у реалізації є Bidirectional Encoder Representations from Transformers, тобто Модель представлення кодувальника з двонапрямковими трансформерами. У дослідженні [13] описано процес порівняння таких моделей як LSTM, Bidirectional LSTM, CNN-BiLSTM та BERT. Експеримент проводився на датасеті [15] з фальшивими та правдивими новинами від ISOT Research Lab з Канадського Університету Вікторії.

Таблиця 3.4.1 – Таблиця даних [15] зі статі-порівняння [13] методів глибинного навчання на задачі розпізнавання фальшивих новин

Дані	Ориг. Трен.	Незбал. Трен.	Ориг. Валід.	Незбал. Валід.	Ориг. Тест.	Незбал. Тест.
Правдиві	13765	13765	3409	3409	4243	4243
Фальшиві	14969	1497	3775	378	4737	474

Таблиця 3.4.2 – Таблиця показників моделей, що слугували об’єктами дослідження у статті [14]

Модель	Збалансована вибірка				Незбалансована вибірка			
	Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
LSTM	0.9697	0.97	0.97	0.97	0,9780	0,98	0,98	0,98
BiLSTM	0.9710	0.97	0.97	0.97	0,9799	0,98	0,98	0,98
C-BiL	0.9722	0.97	0.97	0.97	0,9791	0,98	0,98	0,98
BERT	0.9874	0.99	0.99	0.99	0,9903	0,99	0,99	0,99

BERT є найкращим методом розпізнавання ІПСО з кількох причин. По-перше, він здатний розуміти контекст тексту завдяки своїй контекстуальній моделі. Це означає, що BERT може враховувати зв'язки між словами в реченні, що сприяє кращому розумінню семантики тексту та виявленню шаблонів ІПСО.

По-друге, BERT має перевагу попереднього навчання на великому корпусі текстів. Це дозволяє моделі засвоїти широкий спектр мовних властивостей та використовувати ці знання для кращого розпізнавання ІПСО. Крім того, BERT використовує контекстуальні вектори слів, що дозволяє розрізняти їх значення залежно від контексту, підвищуючи точність розпізнавання.

3.5 Огляд існуючих комерційних програмних рішень

У наступному розділі будуть розглядатися комерційні засоби розпізнавання ІПСО та фальшивих новин, як підвид психологічного впливу. Буде розглянуто лише інструменти, що мають відкритий доступ для українських користувачів та можуть бути швидко знайдені за потреби, адже у перевірці новин важливим є час.

3.5.1 Бот «Перевірка» від Gwara Media

«Перевірка» - це бот, розроблений компанією Ґвара Медіа, який служить для виявлення фейкових новин [16]. Його основна мета - надати користувачам інструмент, який допоможе перевірити достовірність новин та інформації, що поширюються в медіапросторі. Бот використовується зокрема для аналізу ситуації в Харкові та соціальних змін в Україні [16].

Цей інструмент пропонує користувачам послідовні кроки, які допоможуть їм критично оцінити новини та з'ясувати їх достовірність. Крім того, бот надає корисні поради та рекомендації щодо пошуку додаткової інформації, перевірки джерел та переконливості аргументів. Він допомагає користувачам стати більш свідомими споживачами новин та уникнути поширення недостовірної інформації [16].

Інструмент «Перевірка» використовується для боротьби з поширенням фейкових новин та сприяє підвищенню рівня медійної грамотності серед користувачів. Він дозволяє перевірити факти, знайти надійні джерела інформації та зробити об'єктивні висновки. «Перевірка» є важливим інструментом для підвищення свідомості та критичного мислення в інформаційному суспільстві [16].

Перевагами цього ресурсу є:

— Якісна перевірка матеріалу: новини ретельно перевіряються на «правдивість» та «критичність», у ресурсу є база новин, що значно полегшує та пришвидшує цей процес.

— Зручний для користування: щоб скористуватися ботом достатньо лише мати на телефоні додаток Telegram [17] та надіслати декілька команд.

— Локалізація: перевірка здійснюється як українською, так і російською та англійською мовами, що дуже зручно.

Недоліки:

— Довгий час перевірки: сторінка ресурсу сповіщає, що процес обробки та перевірки новини може тривати до доби, що є надзвичайно довго, особливо у рамках повномасштабної війни.

— Залученість «експертів»: як вже було згадано раніше, перевірка здійснюється експертами, що призводить до довготривалої перевірки та потенційних похибок через втому або недостатню кількість опрацьованих джерел.

3.5.2 Додаток до браузера «Fake News Chrome Extension» [18]

Цей інструмент є розширенням для веб-браузера Google Chrome, відомим як "Fake News Chrome Extension". Він розроблений для допомоги користувачам виявляти фейкові новини та недостовірну інформацію, яка поширюється в Інтернеті.

Після встановлення розширення, воно інтегрується з браузером і надає користувачам додаткові функціональні можливості. При перегляді новинних статей або веб-сайтів, розширення використовує різні алгоритми та методи, щоб визначити, чи є ця інформація фейковою чи недостовірною.

Fake News Chrome Extension оцінює джерела новин, перевіряє факти, аналізує мову та структуру тексту, а також використовує дані з надійних джерел та баз знань для порівняння та перевірки інформації. Він надає користувачам спеціальні попередження або підказки, якщо виявляє потенційно фейкові новини або сумнівну інформацію.

Fake News Chrome Extension є корисним інструментом для виявлення фейкових новин та підвищення медійної грамотності користувачів. Він допомагає зменшити поширення недостовірної інформації та сприяє формуванню критичного мислення при споживанні новин в онлайн-середовищі.

Переваги інструменту:

1. Функціональність: Якісна перевірка статей на клікбейт, перебільшення, маніпуляції та фабрикування

2. Зручність: цей інструмент інтегрується у робочий браузер користувача та завжди готовий для розпізнавання

3. Доказовість: цей інструмент також рекомендує статі, що спростовують факти, що наводяться у фальшивих новинах.

Цей інструмент має також і недоліки:

1. Локалізація: Розглянутий інструмент не має покриття новин українською мовою, що унеможливлює перевірку новин на українських медіаресурсах.

2. Труднощі встановлення: Незважаючи на зручність у використанні, встановлення розширення для браузеру може бути доволі неочевидною задачею для недосвідчених користувачів браузерів.

3. Платформа: Сьогодні більшість новин споживається користувачами через телефон, розглянутий інструмент натомість працює тільки у десктопній версії браузеру.

4. Аудиторія: База новин для перевірки є обмеженою та американоцентричною, що обмежує значно використання цього інструменту в українських реаліях.

Останній пункт стосується і решти інтернет-ресурсів з перевірки новин на фальшивість, що не належать українському сегменту інтернету.

3.5.3 Порівняння програмних рішень для розв’язання задачі

На сьогоднішній день, в широкому доступі українцям відсутні інструменти, які б забезпечували швидку та бездоганну перевірку наявності ІПСО. Незважаючи на наявність деяких корисних інструментів, таких як бот "Перевірка" від Gwara

Media і додаток до браузера "Fake News Chrome Extension" та інші, їхні обмеження не можна ігнорувати.

Ці інструменти дійсно мають свої переваги. Вони здатні забезпечити якісну перевірку інформації, допомогу у виявленні фейкових новин та недостовірних даних, а також поради щодо перевірки джерел та фактів. Проте, вони також мають свої недоліки.

Перший недолік полягає у тривалості процесу перевірки. Виявлення фейкових новин та перевірка джерел можуть займати багато часу, особливо якщо потрібно залучати експертів або аналізувати багато джерел. Це може вплинути на швидкість реакції та розповсюдження недостовірної інформації.

Другий недолік пов'язаний з обмеженою підтримкою мови. Деякі інструменти можуть бути обмежені підтримкою лише деяких мов, що ускладнює перевірку новин, що публікуються на інших мовах. Це може створювати прогалини в перевірці інформації з різних джерел.

Третій недолік пов'язаний зі встановленням та використанням інструментів. Деякі з них вимагають встановлення додатків або розширень для браузера, що може бути складним для користувачів з обмеженим технічним досвідом. Це може створювати бар'єри для широкого використання інструментів перевірки.

Усі ці недоліки показують, що, незважаючи на наявність деяких корисних інструментів, все ще потрібно працювати над розвитком більш ефективних та доступних засобів перевірки наявності ІПСО. Вдосконалення таких інструментів є важливим завданням для боротьби з поширенням дезінформації та фейкових новин.

3.6 Висновки до розділу

В даний час, коли дезінформація та фейк-новини стають все поширенішими, необхідно мати в своєму розпорядженні якісні інструменти для розпізнавання

інформаційних повідомлень з ознаками систематичної дезінформації (ІПСО). Однак, загальнодоступних рішень з цією метою є обмежена кількість і подекуди вони не можуть використовуватися пересічними українцями. Це вказує на необхідність розробки нових інструментів, які б забезпечували користувачам якісні та швидкі відповіді.

Під час огляду існуючих рішень для розпізнавання ІПСО, було прийнято рішення про використання архітектури BERT для реалізації моделі. BERT (Bidirectional Encoder Representations from Transformers) є однією з передових архітектур у галузі обробки природної мови (NLP). Вона базується на трансформерних моделях і здатна ефективно розуміти контекст тексту, використовуючи багатошарове навчання.

Одним з головних переваг BERT є його комбінований підхід навчання, який включає два етапи: "pre-training" (попереднє навчання) та "fine-tuning" (доналагодження). Під час попереднього навчання модель використовує великі масштаби текстових корпусів, що допомагає їй усвідомити широкий спектр лінгвістичних залежностей. Потім, на основі такого попереднього навчання, модель доналагоджується для конкретних завдань, таких як класифікація тексту або розпізнавання ІПСО. Цей комбінований підхід дозволяє досягти високої точності і ефективності при роботі з текстовими даними.

У порівнянні з іншими алгоритмами, такими як SVM (Support Vector Machines), наївний баєсівський класифікатор або інші розглянуті інструменти, BERT показує кращі результати. Інші інструменти можуть мати обмежену здатність розуміти контекст та виявляти складні залежності у тексті, що може призводити до менш точних результатів. BERT, з іншого боку, завдяки своїй архітектурі та підходу до навчання, може більш ефективно розпізнавати ІПСО та надавати користувачам якісні відповіді.

4 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 BERT та її модифікація RoBERTa

Модель BERT (Bidirectional Encoder Representations from Transformers) є однією з найпотужніших моделей мовного моделювання в обробці природньої мови (NLP). Ця модель базується на архітектурі трансформерів і відома своїм здатністю до контекстного розуміння мови. BERT вирізняється тим, що вона може одночасно моделювати лівий та правий контексти слова, що дозволяє зрозуміти слово в контексті речення.

Головна інновація BERT полягає в тому, що під час попереднього навчання модель використовує маскування та наступне речення як об'єктиви. Маскування (Masked Language Model, MLM) використовується для навчання моделі передбачати приховані слова в реченні. Деякі слова вхідного речення випадково маскуються, а модель повинна передбачити, які слова були приховані на підставі контексту. Це дозволяє моделі розуміти залежності між словами та розпізнавати їх семантичні значення.

Також, модель навчається передбачати, чи є два речення насправді послідовними. Для цього використовується механізм передбачення наступного речення (Next Sentence Prediction, NSP). Під час навчання модель отримує два речення, і їй потрібно передбачити, чи слідують вони одне за одним в тексті чи ні

Після попереднього навчання BERT може бути доналаштована або доучена на конкретних завданнях, таких як класифікація тексту, відповіді на запитання, машинний переклад та багато інших. Під час доналаштування BERT використовується процес, що називається "доученням" або файн-тюнингом (fine-tuning), де модель пристосовується до конкретного завдання шляхом тренування на деякому обмеженому наборі даних.

BERT демонструє вражаючу продуктивність на багатьох завданнях обробки мови і стала однією з найуспішніших моделей в цій галузі. Її здатність розуміти

контекст і виявляти семантичні залежності між словами зробила її корисною для широкого спектра застосувань, включаючи веб-пошук, аналіз соціальних медіа, автоматичний переклад та інші завдання обробки мови.

RoBERTa (Robustly Optimized BERT approach) є однією з модифікацій моделі BERT, яка була запропонована з метою покращити її продуктивність та розширити її можливості. Одна з основних відмінностей між RoBERTa і BERT полягає в стратегії навчання.

Під час попереднього навчання RoBERTa використовує більше даних і триваліше тренується. Вона також використовує більші партії даних та більш складні маскування. Ці оптимізації допомагають RoBERTa отримувати кращі результати на різних завданнях обробки мови.

Ще одна важлива відмінність полягає в тому, що RoBERTa не використовує об'єктив наступного речення, як це робить BERT. Це дозволяє моделі уникнути деяких шаблонних підходів до розуміння мови, що поліпшує її здатність розрізняти різні семантичні залежності та відносини між словами.

Значущим також ще є те, що RoBERTa використовує динамічно змінювані шаблони маскування під час попереднього навчання. Це означає, що модель може бачити різні комбінації маскованих слів на різних етапах навчання. Такий підхід дозволяє моделі краще усвідомлювати контекстуальні залежності та поліпшує її здатність доуточнювати відсутні слова в реченні.

Окрім того, RoBERTa використовує довші послідовності для тренування, що дозволяє моделі краще усвідомлювати контекст і залежності між словами.

Узагальнюючи, RoBERTa є покращеною версією BERT, яка досягає кращих результатів на різних завданнях обробки мови. Вона використовує оптимізовані стратегії тренування, більше даних та більш складні маскування, що дозволяє їй краще розуміти контекст і виконувати завдання обробки мови з високою точністю.

4.2 Архітектура RoBERTa

RoBERTa використовує Transformer - механізм уваги, який навчається розуміти контекстуальні зв'язки між словами (або підсловами) у тексті. У своїй стандартній формі Transformer складається з двох окремих механізмів - енкодера, який читає вхідний текст, та декодера, який генерує передбачення для завдання. Оскільки RoBERTa спрямована на генерацію мовної моделі, достатньо використовувати лише механізм енкодера.

У відміну від напрямних моделей, які читають вхідний текст послідовно (зліва направо або справа наліво), кодувальник Transformer читає всю послідовність слів одразу. Тому його можна назвати бідирекціональним або двонапрямним, але більш точною є його ненапрямленість. Ця особливість дозволяє моделі засвоювати контекст кожного слова на підставі його оточення (зліва та справа).

RoBERTa має дві основні складові: кодувальник (encoder) та завдання підгодування (pretraining task) (Див. Рисунок 4.2.1). На Рисунку 4.2.2 зображено загальну архітектуру моделі для задач класифікації.

Кодувальник RoBERTa складається з повторюваних блоків, кожен з яких включає багатоголовну увагу (Multi-Head Attention) для моделювання залежностей між словами у тексті. Це дозволяє моделі розглядати контекст зліва та зправа від кожного слова, що сприяє кращому розумінню семантики та значення слів. Крім того, в кодувальнику використовується позиційне кодування, яке допомагає моделі розрізняти слова за їх позиціями у реченні. Також включений шар зворотного зв'язку, який допомагає моделі виявляти складні залежності.

Завдання підгодування BERT полягає у навчанні моделі на великому корпусі текстів за допомогою завдання маскування мови та передбачення наступного речення. Завдання маскування мови включає випадкове маскування деяких слів у вхідному реченні, а потім модель тренується для передбачення цих маскованих слів.

Завдання передбачення наступного речення визначає, чи є два речення суміжними в оригінальному тексті.

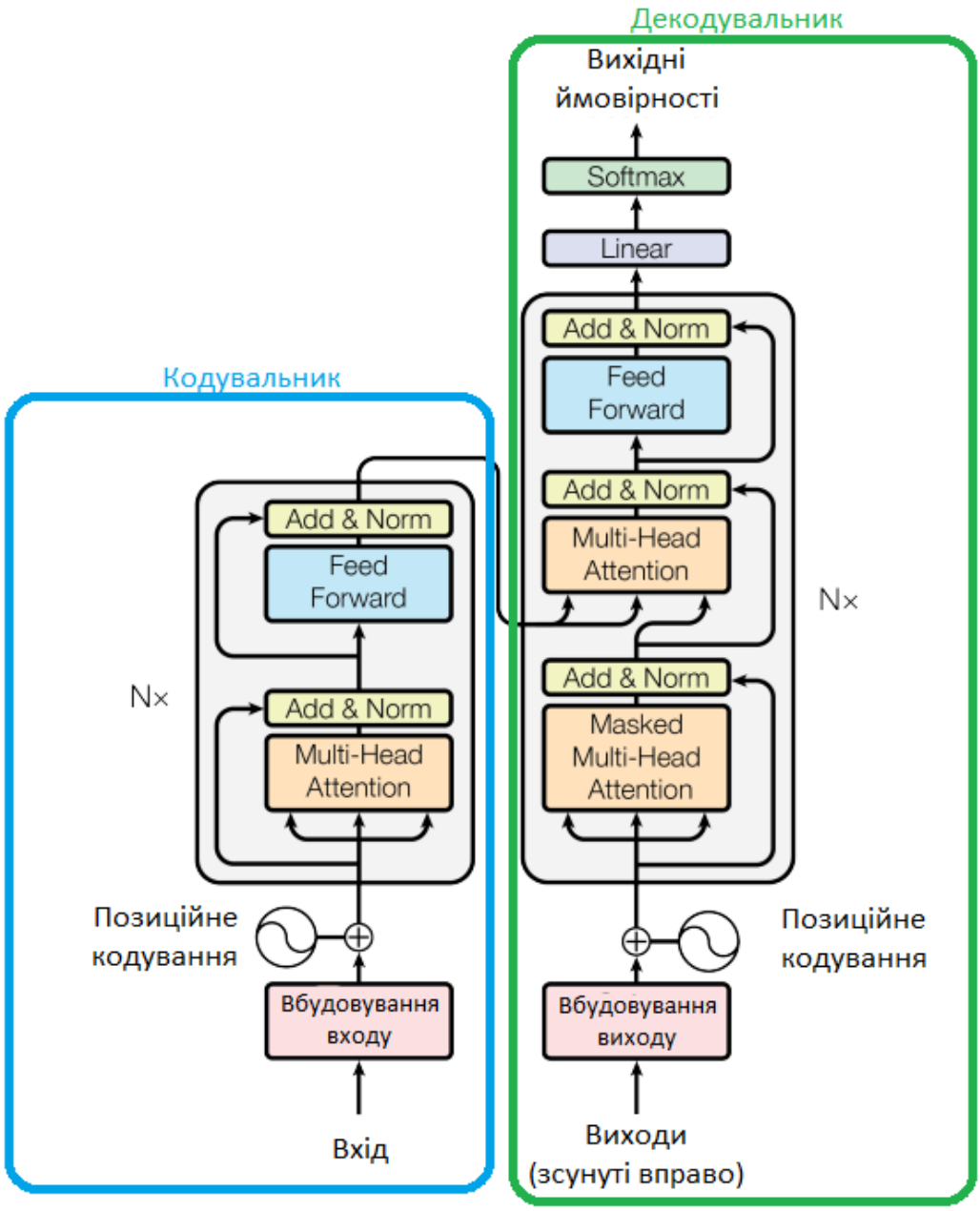


Рисунок 4.2.1 – Зображення архітектури Трансформера [19]

.

Після підгодування на великому корпусі текстів, RoBERTa може бути доналаштований для конкретних завдань NLP, таких як класифікація тексту, позначення іменованих сутностей, відповіді на запитання та інші. Це дозволяє

використовувати модель BERT для широкого спектру природномовних завдань з високою точністю.

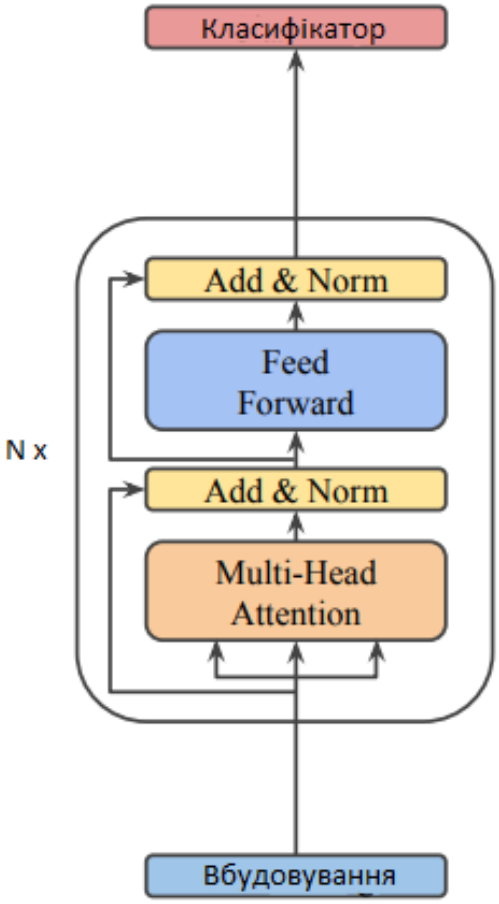


Рисунок 4.2.2 – Зображення архітектури RoBERTa для задач класифікації [23]

4.2.1 Вбудовування (Embedding)

Вбудовування (або ембедінг) в є методом представлення даних, в якому кожен об'єкт, такий як слово, токен або об'єкт, представляється у вигляді вектора числових значень. Це представлення даних здебільшого здійснюється у просторі низької розмірності, що дозволяє моделям машинного навчання працювати з ними ефективно.

Вбудовування використовуються для того, щоб упорядкувати та компактно представити даний об'єкт у векторному вигляді. Вони допомагають зменшити розмірність простору даних, зберігаючи важливу інформацію про об'єкти та їх взаємозв'язки.

Процес створення вбудовувань полягає у навчанні моделі на великому об'ємі даних. Під час навчання модель аналізує взаємозв'язки між об'єктами та намагається знайти унікальні характеристики кожного об'єкта, які потім відображаються у вбудовуванні.

Вбудовування можуть бути створені для різних типів об'єктів, таких як слова в тексті, токени, зображення, аудіофайли та інші. Наприклад, вбудовування слів використовуються для представлення слова у векторному вигляді, де подібні слова мають близькі вектори, що відображає їх семантичну схожість (Рисунок 4.2.1.1).

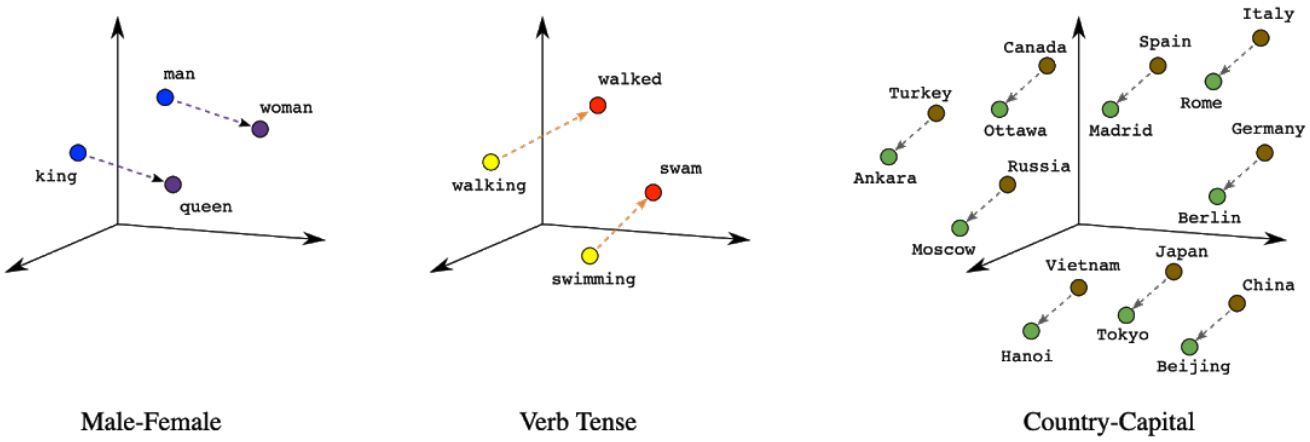


Рисунок 4.2.1.1 – Вектори слів у 3-мірному просторі, зазвичай вбудовані слова можуть мати сотні або тисячі просторів [22]

Вбудовування є потужним інструментом в області машинного навчання та обробки природної мови. Вони дозволяють моделям розуміти семантику та зв'язки між об'єктами, здійснювати класифікацію, кластеризацію, рекомендації та інші завдання, що вимагають аналізу та порівняння даних.

В контексті моделі RoBERTa, вбудовування використовуються для представлення слів або токенів у тексті у векторному вигляді, який може бути оброблений нейромережею. Вбудовування в RoBERTa є однією з перших частин моделі та відповідають за перетворення текстових токенів у числові вектори, які можуть бути подані на вхід нейромережі для подальшої обробки.

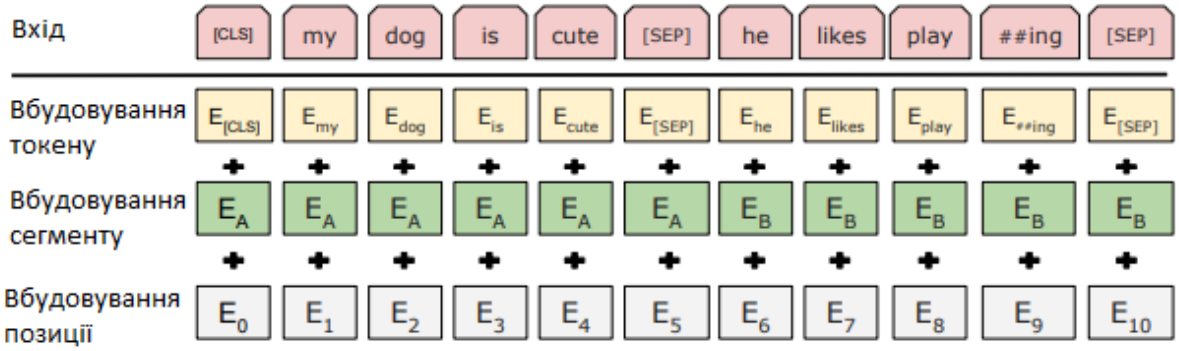


Рисунок 4.2.1.2 – Схема вводу до моделі RoBERTa [20]

У моделі RoBERTa використовується розмір вбудовувань 768, проти 512, що застосовано у BERT. Це означає, що кожен токен або слово у тексті представляється у векторному просторі розміром 768. Більший розмір вбудовувань дозволяє моделі зберігати більше деталей та інформації про кожен токен, що допомагає покращити якість та точність моделі у завданнях обробки природної мови.

Процес створення вбудовувань у RoBERTa включає два основних кроки: токенізацію та векторне представлення.

Токенізація полягає в розбитті вхідного тексту на окремі токени або слова. Наприклад, речення "Я люблю грати в футбол" може бути токенізоване на токени "Я", "люблю", "грати", "в", "футбол".

Далі, кожен токен перетворюється у векторне представлення за допомогою вбудовувального шару. Цей шар є матрицею, де кожному токену відповідає вектор з числовими значеннями. Ці вектори навчаються під час процесу навчання моделі RoBERTa і відображають семантичні та синтаксичні властивості токенів. Важливою

особливістю вбудовувань є те, що вони враховують не тільки сам токен, але й його позицію у вхідному тексті. Це досягається за допомогою додавання до векторного представлення токену позиційного кодування або вбудовування позиції (positional encoding, positional embedding), яке вказує на позицію токену у послідовності (Рисунок 4.2.1.2).

Таким чином, вбудовування у RoBERTa дозволяють представити кожен токен у тексті у векторному вигляді з урахуванням його семантичного значення та позиції. Це дозволяє моделі RoBERTa розуміти контекст та взаємозв'язки між токенами при виконанні завдань обробки природної мови,

4.2.2 Багатоголівна увага (Multi-Head Attention)

Багатоголівна увага є ключовою складовою архітектури трансформерів, яка використовується для моделювання взаємодії між словами або токенами у вхідному тексті. Вона дозволяє моделі фокусуватись на різних аспектах контексту та знаходити важливі залежності між словами.

У блоці багатоголівної уваги вхідний вектор представлення слів розбивається на кілька груп, відомих як "голови" (heads). Кожна головка використовує окремий набір параметрів для обчислення ваги уваги(attention weight) між словами. Це означає, що кожна голова може зосередитись на різних аспектах взаємодії між словами.

У процесі обчислення, для кожної голови виконується наступні кроки:

1. Застосовується лінійне перетворення до вхідного вектору, щоб отримати запит, ключ та значення (query, key, value).

2. Обчислюється оцінка уваги між запитом та ключем, в результаті чого отримується ваговий коефіцієнт, який відображає важливість кожного слова для даного запиту.

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (4.2.2.1)$$

де Q – запит, K – ключ та V – значення. Функція softmax представляється у вигляді рівняння 4.2.2.2

3. Використовуючи ваговий коефіцієнт, обчислюється n-зважена сума значень, що відповідає значенням та вагам.

$$S(y)_i = \frac{\exp(y_i)}{\sum_{j=1}^{n}\exp(y_j)} \quad (4.2.2.2)$$

4. Отримані зважені значення групуються разом та проходять через фінальний лінійний шар для отримання вихідного представлення головки [19].

Потім вихідні представлення голів об'єднуються та проходять через ще один лінійний шар, щоб отримати остаточне представлення Multi-Head Attention(4.2.2.3).

$$MultiHead(Q,K,V) = Concat(head_{\overline{1..h}})W^O \quad (4.2.2.3)$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (4.2.2.4)$$

де W – вагові матриці.

Multi-Head Attention дозволяє моделі віддавати увагу різним аспектам взаємодії між словами, таким як семантика, граматика, порядок слів тощо. Це допомагає покращити якість моделі та її здатність розуміти контекст у вхідних даних.

У гіперпараметрах уваги в архітектурі трансформерів є декілька ключових параметрів, які визначають спосіб обчислення уваги між словами або токенами. Основні гіперпараметри уваги включають наступні:

— Кількість голів (num_heads): Цей параметр визначає, на скільки груп розбивається вхідний вектор для обчислення уваги. Кожна голівка фокусується на різних аспектах контексту. Зазвичай використовуються значення від 8 до 16 головок.

— Розмір простору ключів/запитів/значень (key/query/value dimensions): Ці параметри визначають розмір векторів, що використовуються для представлення ключів, запитів та значень у обчисленні уваги. Зазвичай використовуються значення від 64 до 512.

— Розмір шару уваги (attention layer size): Цей параметр визначає розмір проміжного шару, який застосовується перед обчисленням уваги. Він впливає на складність моделі та кількість обчислень. Зазвичай використовуються значення від 128 до 1024.

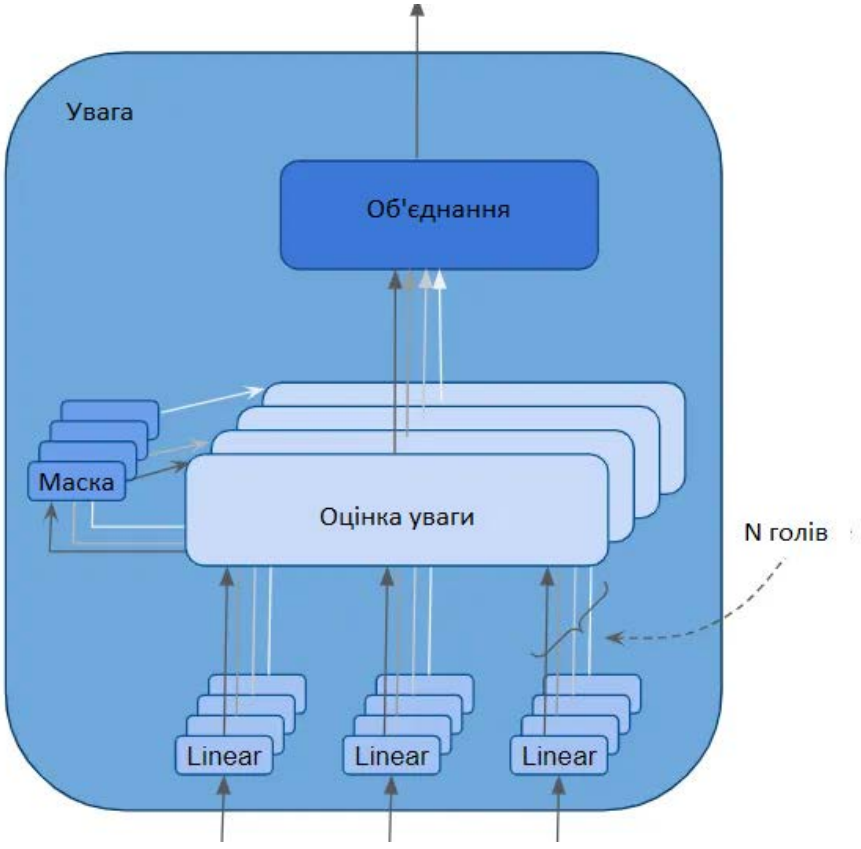


Рисунок 4.2.2.2 – Будова багатоголівної уваги [23]

— Швидкість згасання (dropout rate): Dropout - це техніка регуляризації, яка випадково "відключає" певні нейрони під час навчання для запобігання перенавчанню. Швидкість згасання визначає ймовірність відключення нейронів. Зазвичай використовуються значення від 0.1 до 0.5.

— Функція активації (activation function): Цей параметр визначає функцію, яка застосовується після обчислення уваги. Популярними варіантами є ReLU (Rectified Linear Unit) та GELU (Gaussian Error Linear Unit).

4.2.3 Додавання та нормалізація (Add&Norm)

Add&Norm - це шар, що використовується в архітектурі кодеру. Цей шар виконує дві основні операції: додавання (Add) і нормалізацію (Norm).

Суть Add&Norm шару полягає в тому, що він додає результати вхідного сигналу (наприклад, вихід з попереднього шару) і пропускає їх через процес нормалізації. Це дозволяє зробити модель стійкішою до затухання градієнту під час навчання.

Процес додавання виконується покомпонентно, що означає, що кожен елемент вхідного сигналу додається до відповідного елементу результату. Це дозволяє зберегти інформацію з попереднього шару та врахувати її в подальшому обчисленні.

Після додавання вхідного сигналу до результату застосовується процес нормалізації, який має на меті стандартизувати значення векторів, щоб полегшити подальшу обробку моделлю. Нормалізація здійснюється шляхом віднімання середнього значення вектора і поділу на його середнього крадратичного відхилення.

Така комбінація додавання та нормалізації дозволяє забезпечити стабільність і ефективність нейромережі, полегшує процес навчання та забезпечує швидку

збіжність. Крім того, це також допомагає запобігти проблемі зникнення та вибування градієнту, яка може виникати при навчанні глибоких нейромереж. Також процес додавання допомагає моделя запам’ятовувати позиційні кодування.

Позначимо X та Y вхідними елементами шару нормалізації та додавання, розмірність кожного $k \times m$. Тоді операції, що будуть здійснюватися можна описати формулами 4.2.3.1-4.2.3.4.

$$S = X + Y \tag{4.2.3.1}$$

$$\mu = \frac{1}{k \times m} \sum_{i=1}^{k} \sum_{j=1}^{m} S_{ij} \tag{4.2.3.2}$$

$$\sigma^2 = \frac{1}{k \times m} \sum_{i=1}^{k} \sum_{j=1}^{m} (S_{ij} - \mu)^2 \tag{4.2.3.3}$$

$$Z_{ij} = \frac{S_{ij} - \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{4.2.3.4}$$

де X – вхідні дані попереднього шару, Y – вихідні дані попереднього шару, μ – математичне сподівання, σ^2 – середнє квадратичне відхилення величини, Z_{ij} – елемент нормалізованої матриці.

4.2.4 Пряме поширення (Feed Forward)

Шар прямого поширення в кодувальнику моделі RoBERTa використовується для забезпечення нелінійності та додаткової обробки вхідних сигналів. Він відіграє

важливу роль у витягуванні та перетворенні інформації, що проходить через кодувальник.

Feed-forward шар допомагає моделі виявляти складні залежності між різними словами або токенами в тексті. Він дозволяє моделі виконувати більш складні обчислення та здійснювати нелінійні перетворення для отримання кращих репрезентацій вхідних даних.

Шар прямого поширення складається з двох повнозв'язних шарів (fully connected layers) з активацією між ними. Ці шари мають різні ваги та зсуви (biases). Вхідні вектори з попереднього шару пропускаються через цей шар, після чого вони проходять активаційну функцію, яка дозволяє моделі вводити нелінійність у обробку даних. Активаційною функцією для шарів є ReLu.

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2 \qquad (4.2.4.1)$$

Feed-forward шар також допомагає у витягуванні значущих ознак з вхідних сигналів та стисненні інформації до більш низькорозмірного простору. Він дозволяє моделі концентруватись на ключових аспектах даних та використовувати їх для подальшого обчислення та передачі відповідних інформаційних репрезентацій.

Крім того, використання шару прямого поширення допомагає моделі адаптуватись до різних завдань та контекстів. Вхідні сигнали можуть бути перетворені та агреговані відповідним чином для кращого представлення даних залежно від поставленої задачі.

Шар прямого поширення використовується у кожному блоку кодувальника RoBERTa, допомагаючи моделі ефективно виявляти, узагальнювати та перетворювати інформацію в процесі обробки тексту. Він входить до складу шарів, які забезпечують глибокі та потужні можливості моделі BERT в розумінні природної мови та виконанні завдань обробки тексту.

4.3 Алгоритм попереднього навчання RoBERTa

Попереднє тренування RoBERTa є важливою фазою в розробці цієї потужної моделі обробки природної мови. Цей процес дозволяє моделі засвоїти широкий спектр мовних особливостей та контекстуальних залежностей, які потім можна використовувати для різних завдань розуміння та генерації тексту.

Основна ідея попереднього тренування RoBERTa полягає в тому, щоб навчити модель розуміти мову на основі великої кількості непромаркованих текстових даних. Цей процес складається з двох основних задач: маскування мови та передбачення наступного речкення.

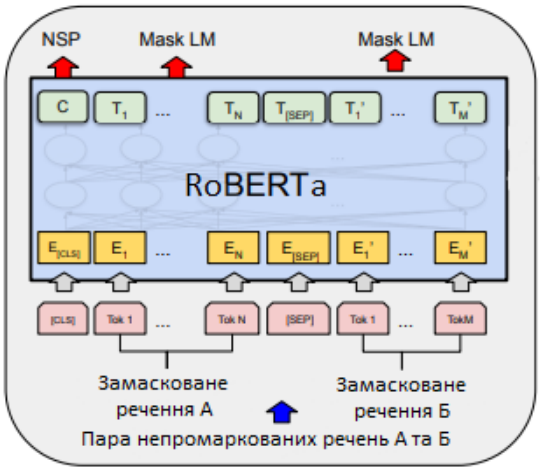


Рисунок 4.3.1 – Процедура попереднього тренування RoBERTa [20]

Обидві задачі попереднього навчання використовують великі обсяги тексту для тренування моделі. Маскування мови дозволяє моделі вчитися враховувати контекст і залежності між словами, тоді як передбачення наступного речення сприяє розумінню семантичних та логічних зв'язків між реченнями. Ці задачі допомагають моделі засвоїти широкий спектр знань про мову та забезпечують підґрунтя для подальших завдань обробки природної мови.

4.3.1 Маскування мови (Masked Language Modeling, MLM)

Задача маскування мови (Masked Language Modeling, MLM) є однією з ключових задач попереднього навчання моделі, зокрема, в контексті RoBERTa. Ця задача полягає в передбаченні правильних значень маскованих токенів у вхідному тексті.

У процесі маскування мови деякі токени випадковим чином маскуються, що означає їх заміну на спеціальний токен [MASK]. Інші токени в тексті можуть також бути масковані або замінені на інші токени, щоб забезпечити різноманітність тренувальних прикладів.

Модель BERT, а також RoBERTa, отримують вхідний текст з маскованими токенами і намагаються передбачити правильні значення цих маскованих токенів. Така задача вимагає враховування контексту всього речення та залежностей між словами. Модель повинна розуміти контекст та семантику речення, щоб впевнено вгадати правильні значення маскованих токенів.

Маскування мови дає можливість моделі засвоїти глибокі залежності та відношення між словами в тексті, а також вчитися розрізняти семантичні та синтаксичні властивості мови. Правильне передбачення маскованих токенів відображає здатність моделі розуміти мову та генерувати адекватні відповіді на основі контексту.

Подивимося на роботу маскувальника на основі речення «Кожного дня я прокидаюся та випиваю склянку води». Для прикладу замаскуємо перше слово, тобто на вхід моделі подамо таке речення: «<mask> дня я прокидаюся та випиваю склянку води.». Результат зображений на Рисунку 4.3.1.1.

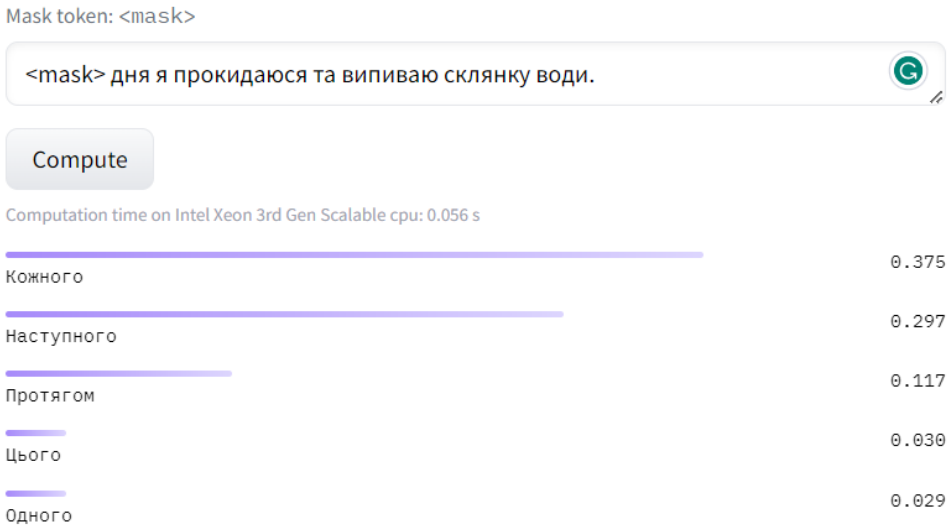


Рисунок 4.3.1.1 – Результат маскування слова «Кожного» моделлю RoBERTa

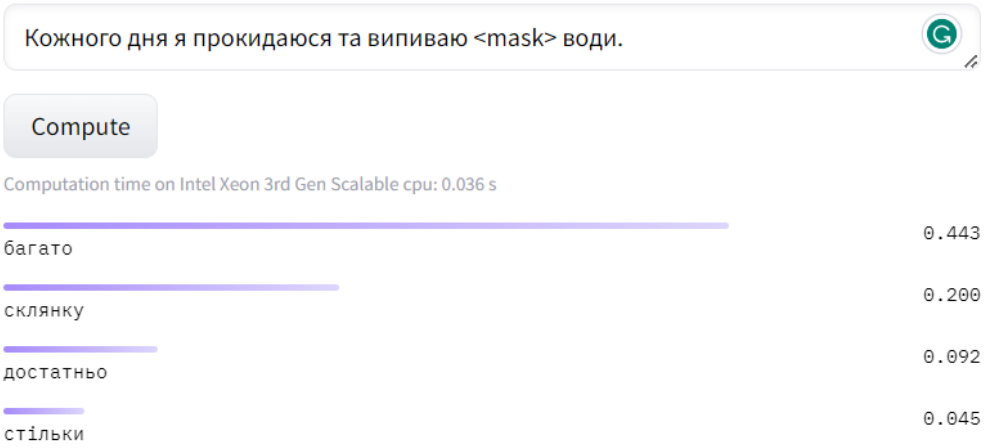


Рисунок 4.3.1.1 – Результат маскування слова «склянку» моделлю RoBERTa

4.3.2 Передбачення наступного речення (Next Sentence Prediction, NSP)

Механізм передбачення наступного речення (Next Sentence Prediction, NSP) є ще однією задачею попереднього навчання, яку використовують моделі, такі як BERT і RoBERTa. Ця задача спрямована на розуміння залежностей між двома реченнями в тексті.

У процесі попереднього навчання, пара речень випадковим чином обирається з корпусу текстів. Одне речення стає "реченням А" (Sentence A), а інше - "реченням В" (Sentence B). Модель отримує цю пару речень в якості вхідних даних.

Метою механізму NSP є передбачення, чи є "речення В" фактично наступним реченням після "речення А" в оригінальному тексті. Модель отримує вхідні вкладення речень, обробляє їх у своєму внутрішньому контекстуальному представленні та намагається визначити, чи вони мають послідовний зв'язок.

Для цього модель може використовувати різні стратегії. Наприклад, вона може обробляти "речення А" та "речення В" окремо і згодом виконувати порівняння на основі їх внутрішнього представлення. Модель також може використовувати підсумковий вектор, отриманий після обробки обох речень, для передбачення ймовірності зв'язку між ними.

Механізм передбачення наступного речення надає моделі контекстуальну інформацію про послідовність речень у тексті. Це допомагає моделі розуміти довгострокові залежності між реченнями та використовувати цю інформацію для кращого розуміння тексту в цілому.

4.4 Алгоритм спеціалізованого навчання або профільованого довчання RoBERT (файн-тюнинг)

У задачі профільованого тренування для моделі RoBERTa, використовуються певні шари, які служать для класифікації і розв'язання конкретної задачі. Ці шари включають лінійне перетворення, функцію активації, випадкове виключення деяких зв’язків та ще одне лінійне перетворення.

Першим шаром є лінійне перетворення, яке змінює розмірність вхідних даних, допомагаючи моделі виділяти корисну інформацію. Далі застосовується функція

активації, яка нелінійно обробляє вихідні значення шару, допомагаючи моделі вчитися складнішим залежностям.

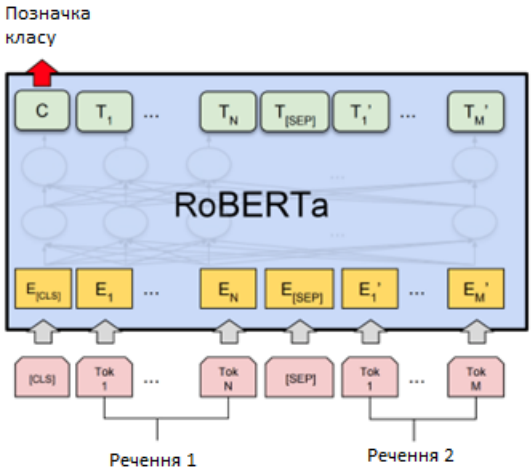


Рисунок 4.4.1 – Процедура спеціалізованого навчання RoBERTa [21]

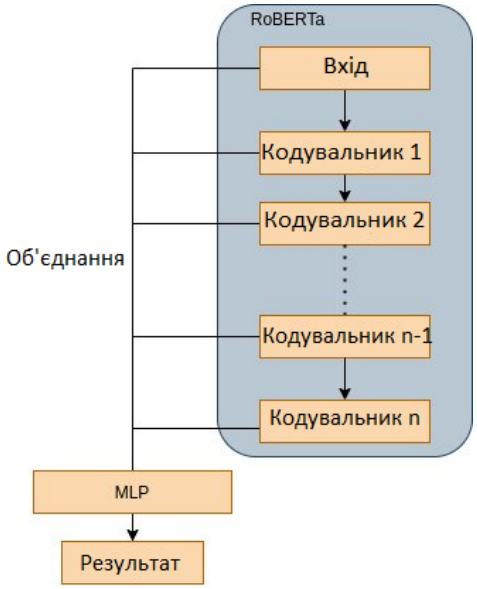


Рисунок 4.4.2 – Архітектура класифікатора на базі RoBERTa

Шар випадкового виключення (dropout) використовується для запобігання перенавчанню. Він випадково "виключає" деякі зв’язки з попереднім шаром, що допомагає моделі краще узагальнювати на нових даних.

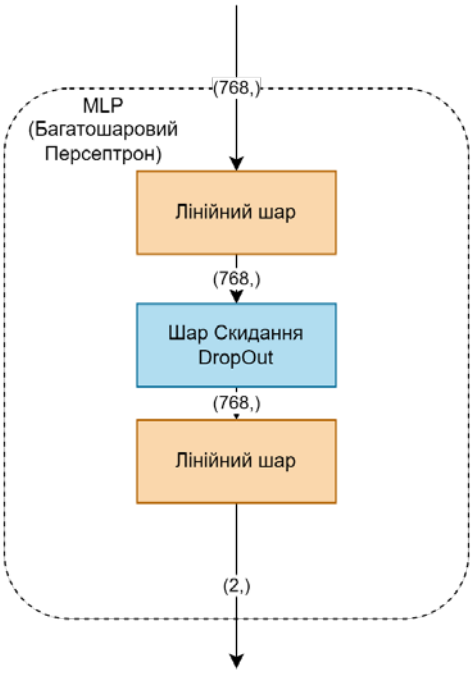


Рисунок 4.4.3 – Зображення будови класифікатора

Останній шар є лінійним перетворенням, яке перетворює проміжний розмірний простір на кінцевий простір, в якому кожен нейрон відповідає окремому класу. Цей шар виконує фінальну класифікацію і використовується для передбачення ймовірності належності вхідного тексту до кожного класу.

Ці шари разом створюють класифікатор, який може бути використаний для розв'язання специфічних завдань класифікації.

4.5 Висновки до розділу

Математичне забезпечення моделі RoBERTa є важливою складовою, яка допомагає моделі розуміти та обробляти природну мову. RoBERTa використовує архітектуру трансформера, що слугує основою для обробки тексту. Трансформер базується на механізмі уваги, який дозволяє моделі встановлювати контекстуальні

залежності між словами у тексті. Він складається з кількох шарів, які обмінюються інформацією, враховуючи взаємодію між різними словами. Це допомагає моделі отримувати багатофакторний контекст та краще розуміти семантичні зв'язки у тексті.

Другий важливий аспект - вбудування (embedding). Вбудування перетворює слова або токени у тексті на числові вектори. Кожному слову або токену ставиться у відповідність числовий вектор, що дозволяє моделі працювати з текстом у вигляді числових значень. Вбудування розміщуються на початку моделі та передаються в трансформер для подальшої обробки.

Задача маскування мови є ще одним важливим елементом математичного забезпечення. Під час попереднього навчання модель у випадковому порядку маскує деяку частину слів у тексті і намагається передбачити їх. Це навчає модель ураховувати контекст та залежності між словами, що допомагає їй краще розуміти текст та виявляти семантичні відношення.

Окрім цього, RoBERTa проходить довгий процес попереднього навчання на великих наборах даних. Під час цього навчання використовуються дві задачі - маскування мови та передбачення наступного речення. Перша задача полягає у передбаченні прихованих слів або токенів у тексті, тоді як друга задача полягає в передбаченні, чи йде одне речення після іншого в оригінальному тексті. Ці задачі допомагають моделі засвоїти широкий спектр знань про мову та структуру тексту. Всі ці математичні компоненти працюють разом, надаючи моделі RoBERTa потужні здібності розуміння та аналізу природної мови. Вони дозволяють моделі ефективно обробляти текст, встановлювати залежності між словами та здійснювати якісні прогнози для різноманітних завдань обробки мови.

5 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

5.1 Опис даних

У цьому дослідженні для тренування та тестування моделі були використані дані, які були зібрані з двох джерел. Перше джерело - відкритий датасет "Propaganda and Fake News in the War in Ukraine" з періоду з 01.02.2022 по 26.04.2022. Цей датасет містить новини, які включають у себе пропаганду. Друге джерело - відкритий датасет "Ukrainian News", який включає в себе новини з різних веб-сайтів та Telegram-каналів.

Обидва джерела дали можливість отримати значну кількість даних для дослідження. В результаті зібрано більше 36 тисяч новин, які включають у себе телеграм-пости та статті з новинних ресурсів. Важливо відзначити, що дані з першого джерела були відібрані як зразок пропаганди та інформаційно-психологічної операції, тоді як дані з другого джерела представляють загальні новини, з яких було відібрано інформацію з надійних джерел.

Використання даних з різних джерел надає більш широку перспективу при аналізі та розробці моделі. Це дозволяє охопити різні аспекти новинного контенту та побудувати більш репрезентативну модель, що допомагає досягти більш точних та надійних результатів.

5.1.1 Джерело правдивих новин

Датасет "Українські новини" , доступний на Hugging Face, є колекцією новинних статей, зібраних з різних українських веб-сайтів та каналів у Telegram.

Даний датасет містить 22 567 099 об'єктів у форматі JSON (новин), загальний обсяг становить близько 67 ГБ. Кожна новина в датасеті містить наступні поля:

—Title: Заголовок новини.

— Text: Текст новини, який може містити HTML-теги (наприклад, абзаци, посилання, зображення та інше).

— URL: Посилання на новину.

— Datetime: Дата публікації або дата, коли новина була зібрана та додана до датасету.

— Owner: Назва веб-сайту, який опублікував новину.

Датасет складається з новин, зібраних з веб-сайтів (16 022 416) та повідомлень з каналів у Telegram (6 544 683). JSON-об'єкти розділені на частини, і весь датасет доступний для завантаження через Hugging Face.

Важливо відзначити, що умови використання датасету зазначають, що всі дані в цьому датасеті підпадають під авторські права власників відповідних веб-сайтів. Тому важливо звертатись до окремих веб-сайтів для отримання додаткової інформації про їх умови використання.

5.1.2 Джерело пропагандистських новин

Датасет "Propaganda and Fake News in the War in Ukraine" [24] є відкритим набором даних, який містить інформацію про пропагандистські та фейкові новини, пов'язані з війною в Україні. Цей датасет був створений для аналізу та дослідження пропаганди та фейкових новин у контексті конфлікту в Україні.

Датасет містить текстові дані, такі як заголовки та текст новин, які включають пропагандистський або фейковий зміст. Крім тексту, також надається інформація про джерело новин, дату публікації та URL посилання на статтю.

Особливістю цього датасету є те, що він фокусується на аналізі пропагандистських та фейкових новин у контексті війни в Україні. Він може бути

використаний для розробки та оцінки моделей машинного навчання, спрямованих на виявлення та класифікацію пропагандистських або фейкових новин.

Цей датасет має значну кількість зразків новин, що дозволяє проводити широкий аналіз та дослідження проблеми пропаганди та фейкових новин. Використання цього датасету може сприяти кращому розумінню природи та розповсюдження пропаганди та фейкових новин у контексті війни в Україні і сприяти розвитку ефективних методів їх виявлення та боротьби з ними.

Оскільки датасет з пропагандою містив новини, написані російською мовою, для його використання у дослідженні та аналізі було необхідно перекласти ці новини на українську мову. Для цього було використано Google Translate API - сервіс машинного перекладу, розроблений Google.

Google Translate API надає можливість автоматичного перекладу тексту з однієї мови на іншу, використовуючи сучасні методи машинного навчання. Завдяки цьому сервісу, новини з датасету з пропагандою, написані російською мовою, були успішно перекладені на українську.

Переклад тексту з однієї мови на іншу може бути викликом, особливо при роботі зі складними текстами або контекстом, який може вплинути на семантику. Тому важливо мати на увазі, що перекладені новини можуть містити деякі неточності або втрату нюансів, які можуть вплинути на точність та достовірність аналізу.

Застосування Google Translate API у цьому контексті було необхідним кроком для забезпечення доступу до змісту новин з датасету з пропагандою російською мовою та його подальшого використання у контексті українськомовного дослідження та аналізу.

5.1.3 Тренувальний та тестувальний набори даних

У процесі підготовки датасету для тренування та тестування моделі було відібрано більше 18 тисяч новин з надійних джерел, що включають у себе офіційні українські джерела інформації. Ці новини були вибрані з таких джерел, як 'Офіс Президента України', 'МВС України', 'Суспільне', 'Громадське' та інші, оскільки ці джерела вважаються надійними та достовірними.

Також було відібрано понад 18 тисяч новин, що містять інформаційно-психологічні операції (ІПСО). Ці новини були зібрані з датасету, що містить пропагандистський контент та інформацію, спрямовану на маніпуляцію та вплив на свідомість аудиторії

Усі зібрані новини були очищені від порожніх входжень та інших невалідних даних. Потім ці новини були об'єднані у один датасет з двома основними полями: "text" (текст новини) та "label" (позначка), яка вказує, чи є новина достовірною (з надійних джерел) чи містить ІПСО.

Для подальшого використання датасету у процесі тренування та тестування моделі, датасет був розділений на тренувальну та тестову вибірки. Тестова вибірка складає приблизно 33% загального обсягу датасету, що дозволяє оцінити ефективність моделі на незалежних даних.

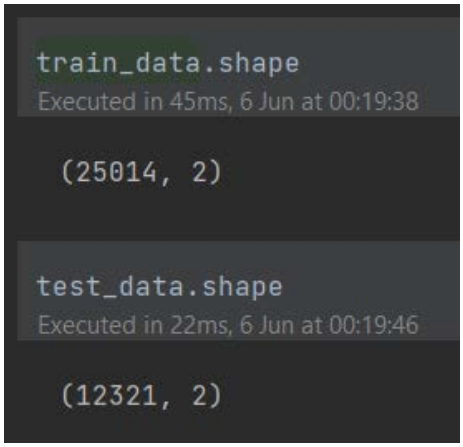


Рисунок 5.1.3.1 – Розміри тренувальних та тестувальних даних

Таким чином, після всіх цих кроків було підготовлено готовий датасет, який може бути використаний для тренування та тестування моделі для класифікації новин на основі їх достовірності та виявлення ІПСО.

Отриманий датасет було опубліковано на платформі Hugging Face. Для доступу до цього датасету достатньо використати функцію load_dataset з бібліотеки Hugging Face, як показано на Рисунку 5.1.3.2.

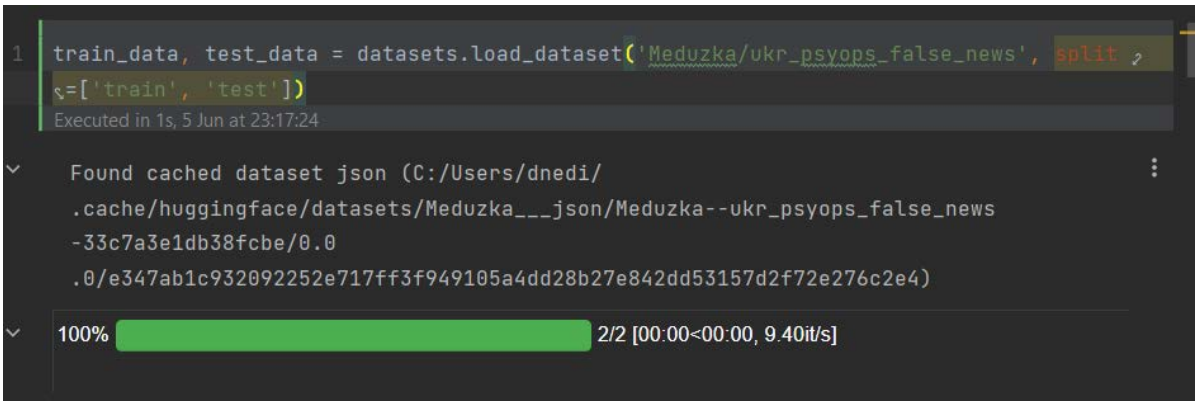


Рисунок 5.1.3.2 – Приклад завантаження датасету з Hugging Face

Цей простий код дозволяє зручно завантажити тренувальні та тестові дані з датасету. Це надає можливість користувачам використовувати цей датасет для власних цілей та робить його дуже зручним та доступним для широкого кола використання.

5.1.4 Формат вхідних даних

Для тренування моделі використовується набір даних, який повинен бути у форматі, сумісному з бібліотекою Hugging Face "datasets". Цей набір даних містить два основних поля: текстове поле, яке містить вхідні дані для моделі, і поле маркування (label), яке вказує на очікувані вихідні значення для цих даних. Формат даних, прийнятий бібліотекою "datasets", забезпечує зручну обробку і підготовку

даних для тренування моделі, дозволяючи ефективно використовувати їх для навчання та оцінки моделі.

Першим кроком перед безпосереднім передбаченням класу є токенізація вхідного тексту за допомогою токенізатора, який побудований на основі тієї ж переднатренованої моделі RoBERTa. Токенізація розбиває текст на окремі токени або підрядки, що представляють окремі слова, символи або сегментацію, згідно з правилами та специфікаціями моделі. Цей процес дозволяє перетворити текстові дані на послідовність числових ідентифікаторів, які модель може розуміти та обробляти. Таким чином, токенізація готує дані для подальшої обробки та передачі їх в модель для передбачення класу.

```
Я люблю токенізувати текст
tensor([    0,   848, 12642, 32422,   296,   456,  1256,  6849,     2])
tensor([1, 1, 1, 1, 1, 1, 1, 1, 1])
Реконструкція:
 ['<s>', 'Ð¯', 'ĠÐ»ÑİÐ±Ð»Ñİ', 'ĠÑĤÐ¾Ð°', 'ÐµÐ¼', 'ÑĸÐ·', 'ÑĥÐ²Ð°ÑĤÐ¸', 'ĠÑĤÐµÐºÑģÑĤ', '</s>']
 <s>Я люблю токенізувати текст</s>


Моя улюблена страва <mask>
tensor([    0, 19848, 42655, 30052,     4,     2,     1,     1,     1])
tensor([1, 1, 1, 1, 1, 1, 0, 0, 0])
Реконструкція:
 ['<s>', 'ÐĽÐ¾Ñı', 'ĠÑĥÐ»ÑİÐ±Ð»ÐµÐ¼Ð°', 'ĠÑģÑĤÑĢÐ°Ð²Ð°', '<mask>', '</s>', '<pad>', '<pad>', '<pad>']
 <s>Моя улюблена страва<mask></s><pad><pad><pad>
```

Рисунок 5.1.4.1 – Приклад токенізованого тексту з та без маскування

5.1.5 Формат вихідних даних

На виході з моделі отримується об'єкт `transformers.modeling_outputs.SequenceClassifierOutput`, який містить різні інформаційні поля, включаючи `logits`. `logits` є тензором, що містить значення афінної функції для кожного класу в задачі класифікації.

```
SequenceClassifierOutput(loss=None,
logits=tensor([[-4.1693,  3.7411]],
grad_fn=<AddmmBackward0>),
hidden_states=None, attentions=None)
```

Рисунок 5.1.5.1 – Приклад вихідних даних

Щоб отримати передбачений клас, необхідно вивести індекс найбільшого елементу в `logits` за допомогою функції аргументу з максимальним значенням, наприклад, `argmax()`. Це дозволяє отримати індекс класу з найбільшою ймовірністю та використати його для подальшого аналізу або прийняття рішень.

```
tensor([[-4.1693,  3.7411]],
grad_fn=<AddmmBackward0>)
```

Рисунок 5.1.5.2 – Приклад тензору вагів для класів, у цьому випадку найбільшим є клас 1, тобто модель передбачила, що дані містять ІПСО.

5.2 Опис моделі

5.2.1 Опис попередньо натренованої моделі

Модель "ukr-roberta-base" є переднатренованою моделлю заснованою на архітектурі RoBERTa, спеціально розробленою для української мови. Вона була створена компанією YouScan з використанням великого обсягу українського тексту з різних джерел.

Модель "ukr-roberta-base" має значну кількість параметрів і була натренована на широкому спектрі завдань обробки мови. Це дозволяє їй розуміти та генерувати природну мову українською. Вона володіє здатністю до семантичного розуміння тексту, розпізнавання відношень між словами та розуміння контексту. Це робить її потужним інструментом для різних завдань обробки мови, включаючи класифікацію, машинний переклад, згорткові генерації тексту та багато інших.

Модель "ukr-roberta-base" доступна для завантаження та використання за допомогою бібліотеки Hugging Face. Це дозволяє розробникам та дослідникам з легкістю використовувати цю модель у своїх проектах та дослідженнях з обробки української мови. Вона надає широкі можливості для покращення результатів обробки тексту та забезпечує швидкий доступ до передових засобів обробки мови для україн Переднатренована модель "ukr-roberta-base" складається з 12 шарів RoBERTa. Кожен шар має 768 прихованих нейронів, що є максимальною кількістю токенів, яку вона може обробляти за один раз. Енкодер моделі містить 12 голів уваги, які дозволяють моделі зосередитися на різних аспектах вхідного тексту і розпізнавати залежності між словами.

Модель "ukr-roberta-base" має вражаючу кількість параметрів - 125 мільйонів. Ці параметри вивчені під час переднатренування моделі на великому обсязі українського тексту. Вони дозволяють моделі засвоїти різноманітні мовні особливості та глибокі зв'язки між словами, що робить її ефективним інструментом для різних завдань обробки мови.

Таблиця 5.2.1.1 – Перелік даних, що були використані для попереднього навчання моделі

Таблиці	Записи	Слова	Символи
Українська Векіпедія – Травень 2020	18 001 466	201 207 739	2 647 891 947
Український OSCAR, дедуплікований	56 560 011	2 250 210 650	29 705 050 592
Записи з соціальний мереж	11 245 710	128 461 796	1 632 567 763

5.2.2 Опис тренування та валідації моделі для задачі класифікації

Для розпізнавання ІПСО (інформаційно-психологічних операцій) було проведено профільне тренування моделі на основі спеціально підготовленого датасету [29]. Цей датасет містить велику кількість новин, які були анотовані відповідно до класифікації ІПСО. Профільне тренування моделі зайняло 6 годин, під час яких модель навчалася розпізнавати патерни та характеристики ІПСО в тексті новин.

Після тренування модель досягла високої точності класифікації, що становить 99%. Це означає, що модель здатна з високою надійністю відрізняти новини, що містять ІПСО, від інших новин. Однак, важливо зазначити, що тренувальні дані містять новини, що походять тільки з періоду з січня до квітня 2022 року. Тому, точність класифікації може знизитися для сучасних новин, оскільки можуть з'явитися нові типи ІПСО або змінитися їх характеристики.

Незважаючи на це, завдяки своїй потужній архітектурі та навчанню на широкому діапазоні даних, модель все ж може успішно справлятися з новинами сьогодення. Вона залишається ефективним інструментом для розпізнавання ІПСО та допомагає виявляти та аналізувати шкідливі впливи інформаційних операцій у текстових матеріалах.

Epoch	Training Loss	Validation Loss	Accuracy	F1	Precision	Recall
0	0.061000	0.040847	0.988962	0.988963	0.988001	0.989927
1	0.000500	0.036472	0.992208	0.992184	0.994451	0.989927
2	0.000000	0.044331	0.993182	0.993171	0.993979	0.992364

```
TrainOutput(global_step=2343, training_loss=0.0436033904151366, metr
d': 8.536, 'train_steps_per_second': 0.267, 'total_flos': 1.97270144
0})
```

```
trainer.evaluate()
```

[1541/1541 06:24]

```
{'eval_loss': 0.03647216036915779,
 'eval_accuracy': 0.9922084246408571,
 'eval_f1': 0.9921836834391792,
 'eval_precision': 0.9944507915782601,
 'eval_recall': 0.9899268887083672,
 'eval_runtime': 390.929,
 'eval_samples_per_second': 31.517,
 'eval_steps_per_second': 3.942,
 'epoch': 3.0}
```

Рисунок 5.2.2.1 – Показники моделі під час першого тренування на 3 епохи

5.3 Опис інтерфейсу

Було вирішено створити телеграм бот як інтерфейс користувача для системи. Цей бот доступний завжди і дозволяє користувачам легко розпізнавати новини та

отримувати загальні рекомендації з інформаційної гігієни, підкріплені надійними джерелами інформації.

Таке рішення є природнім, оскільки багато людей споживають новини з різних телеграм-каналів. Створення такого інструменту допомагає користувачам швидше і зручніше перевіряти достовірність новин. Просто перешліть новину в бота, що займає всього лише 3 кліки: виділити текст новини, натиснути кнопку "Переслати" та вибрати бота серед отримувачів. Цей простий процес дозволяє користувачам швидко та без зусиль перевіряти новини на доброякісність.

Такий телеграм бот є зручним інструментом для користувачів, особливо для тих, хто вже активно використовує телеграм як свій основний месенджер. Він забезпечує швидкий доступ до інформації та допомагає перевіряти новини безпосередньо з обраного месенджера. Це дозволяє користувачам бути більш обізнаними та свідомими споживачами інформації, надаючи їм зручність та надійність у перевірці новинних повідомлень.

5.3.1 Функціонал ПЗ

В телеграм боті вже доступні дві команди для користувачів: "/start" та "/help". Команда "/start" ініціює роботу розпізнавача, який починає функціонувати постійно з моменту його активації. Це означає, що користувачі можуть взаємодіяти з ботом без необхідності перебувати у меню та вводити команди.

Наприклад, якщо користувач відправив команду "/start" під час першої взаємодії з ботом, розпізнавач буде постійно активним, готовим реагувати на будь-які повідомлення, що надійшли в чат. Це дає змогу користувачеві звертатися до бота безпосередньо, навіть якщо він перебуває в іншому вікні або зайнятий іншими справами. Він може просто переслати повідомлення у чат бота, а розпізнавач

обробить його та надасть необхідну інформацію чи виконає відповідні дії відповідно до вхідного повідомлення.

Таким чином, завдяки команді "/start" користувачі можуть зручно та безперервно взаємодіяти з нашим телеграм ботом, незалежно від свого поточного стану або місцезнаходження. Це дозволяє забезпечити ще більш зручний та безперебійний досвід користування нашою системою.

Рисунок 5.3.1.1 – Чат-Бот, команди для роботи з ботом

У нашому телеграм боті ми також включили команду "/help", яка надає користувачам доступ до додаткової інформації. Команда "/help" дозволяє отримати

відповідну допомогу та інструкції з питань взаємодії з ботом та спілкування з розробником системи. Вона надає користувачам інформацію про те, як зв'язатися з нами для отримання допомоги, надання фідбеку або повідомлення про проблему, з якою вони зіштовхнулися.

Крім того, у наших майбутніх планах розробки бота входить новий функціонал у вигляді команди "/information". Цей тег міститиме основні правила інформаційної гігієни та приклади інформаційно-психологічних впливів (ІПСО). Користувачі зможуть отримати доступ до цієї інформації, яка допоможе їм розпізнавати ІПСО та дотримуватися кращих практик інформаційної гігієни.

Завдяки командам "/help" та майбутній команді "/information" ми прагнемо забезпечити користувачам нашого телеграм бота зручний доступ до додаткової інформації та підтримки. Ми вважаємо, що це важливий аспект нашої системи, оскільки дозволяє надати користувачам необхідні знання та допомогу для ефективного використання системи та підвищення рівня інформаційної грамотності.

5.3.2 Сценарій користування

Сценарій 1 – Користувач тестує новину з сайту

Крок 1. Запуск боту командою /start

Крок 2. Копіювання вмісту статті. Інформація має бути не більше 800 слів (Див. Рисунок 5.3.2.1).

Крок 3. Надсилання інформації боту. Отримання результату сканування (Див. Рисунок 5.3.2.2).

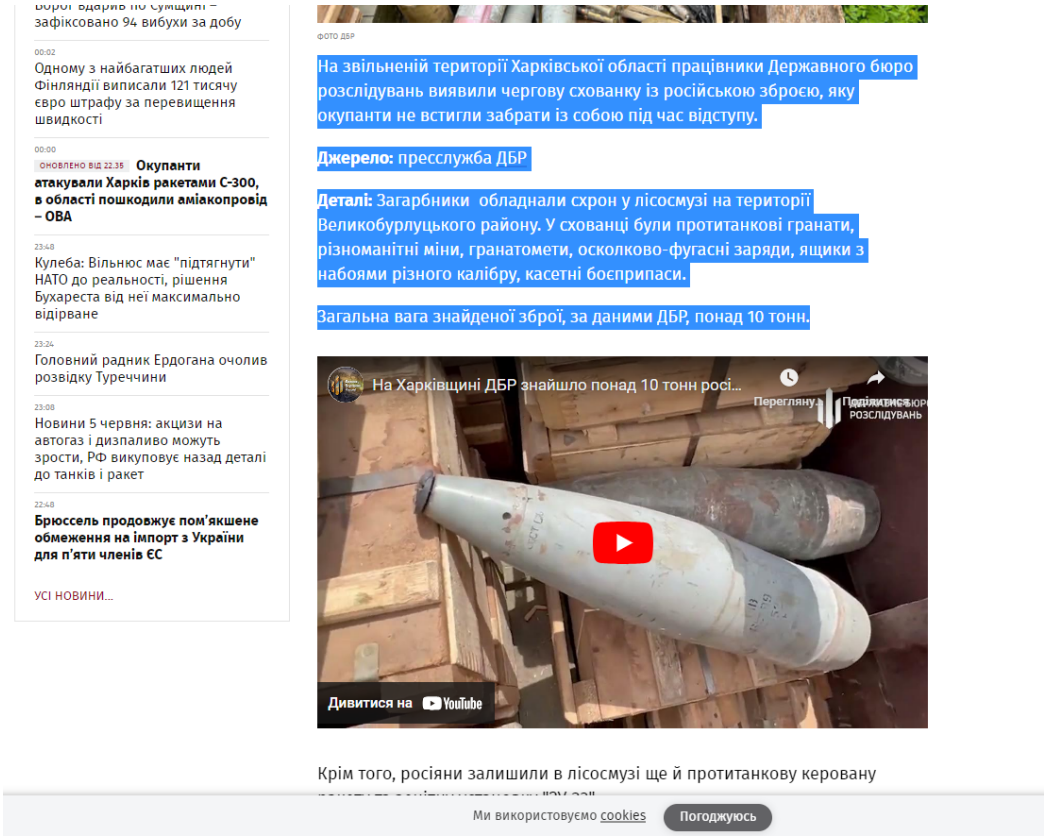


Рисунок 5.3.2.1 – Приклад копіювання новини з інтернет-порталу

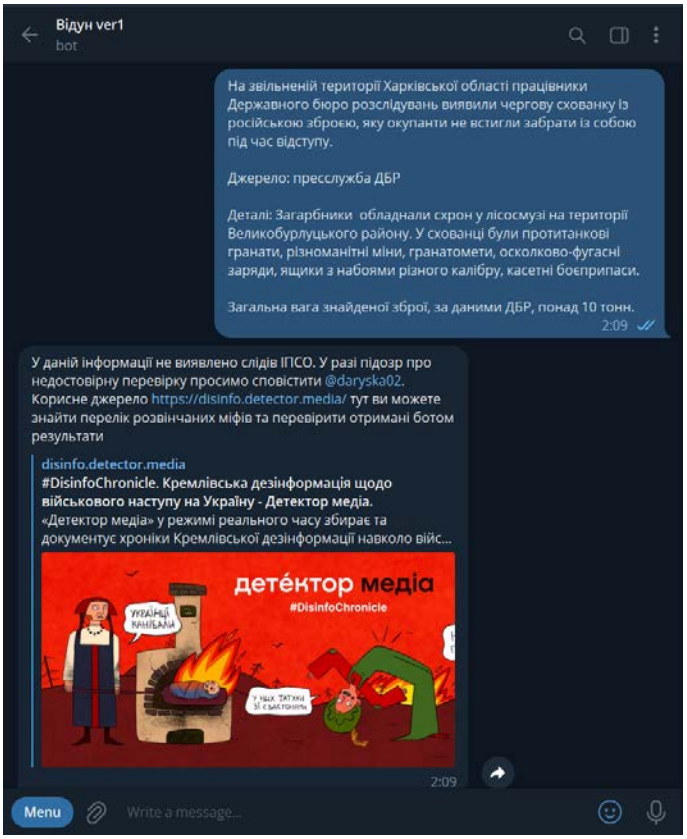


Рисунок 5.3.2.2 – Приклад роботи бота

Сценарій 2 – Користувач тестує новину з телеграм каналу

Крок 1 – Запуск боту

Крок 2 – Копіювання новини у буфер обміну

Крок 3 – Надсилання інформації боку, отримання результату.

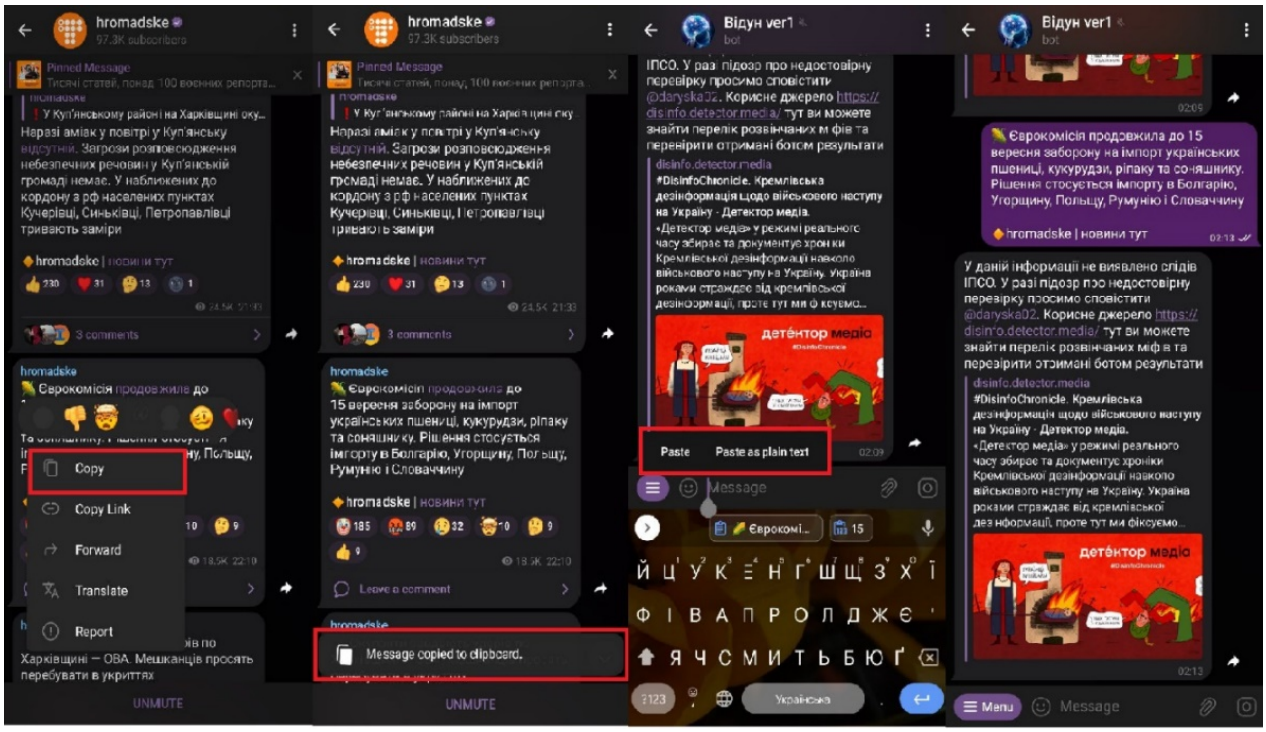


Рисунок 5.3.2.3 – Ілюстрація до сценарію 2, перевірка новини з телеграм каналу

Сценарій 3 – Користувач пересилає новину з каналу у бот

Крок 1 – Запуск боту

Крок 2 – Користувач натискає кнопку переслати поруч з постом

Крок 3 – Користувач обирає бота та натискає «надіслати»

Крок 4 – Користувач отримує відповідь у сповіщенні (Див. Рисунок 5.3.2.4)

Сценарій 4 – Користувач помилково вводить коротке беззмістовне повідомлення

Крок 1 – Користувач запускає бота

Крок 2 – Користувач пише нерозбірливе повідомлення

Крок 3 – Користувач отримує відповідь (Див. Рисунок 5.3.2.5)

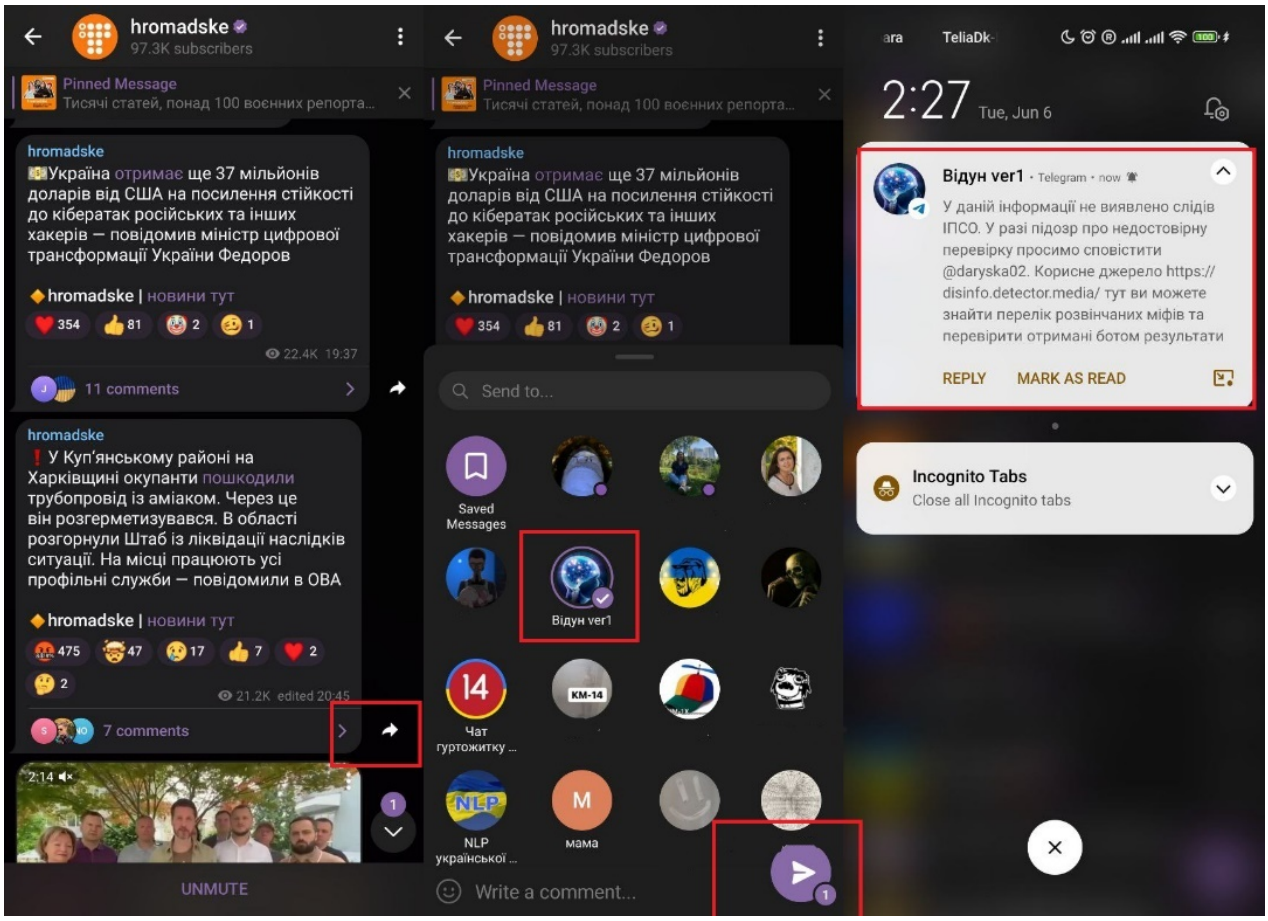


Рисунок 5.3.2.4 Ілюстрація сценарію 3, перевірка пересланої з каналу новини

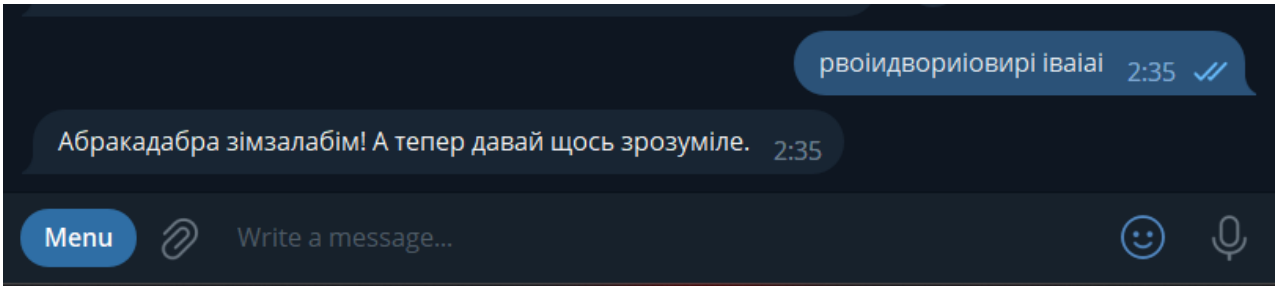


Рисунок 5.3.2.5 Ілюстрація реакції боту на нерозбірливе повідомлення

5.4 Огляд результатів

5.4.1 Показники метрик моделі

Час, витрачений на тренування моделі на протязі п'яти епох, склав 301 хвилину, що є досить тривалим, але відображає велику обчислювальну потужність та складність моделі, такої як RoBERTa. Точність передбачення на тестовій вибірці становить 99.28%, що свідчить про високу ефективність та надійність моделі. Цей результат перевищує очікування і вказує на те, що модель має великий потенціал у виявленні та класифікації ІПСО. Висока точність підтверджує, що модель RoBERTa здатна надійно розпізнавати та відрізняти інформаційні повідомлення, що містять ознаки дезінформації або фейкових новин. Таблиця 4.4.1.1 відображає показники моделі на кожній епосі тренування. Рисунки 4.4.1.2 та 4.4.1.3 відповідно відображають динаміку росту точності та падіння значень функції втрат на кожній епосі тренування.

Таблиця 4.4.1.1 – Таблиця значень метрик моделі під час тренування

Епоха	Тренувальні втрати	Валідаційні втрати	Точність	F1 міра	Влучність (Precision)	Повнота (Recall)
1	0.0376	0.043149	0.99278	0.99277	0.99253	0.993014
2	0.0	0.045322	0.99269	0.99268	0.993	0.992364
3	0.0001	0.069402	0.990423	0.99036	0.99541	0.985378
4	0.0	0.58773	0.99277	0.99277	0.99157	0.993989
5	0.0	0.58675	0.992695	0.99268	0.99365	0.991714

Рисунок 5.4.1.2 – Графік точності моделі на кожній епосі тренування

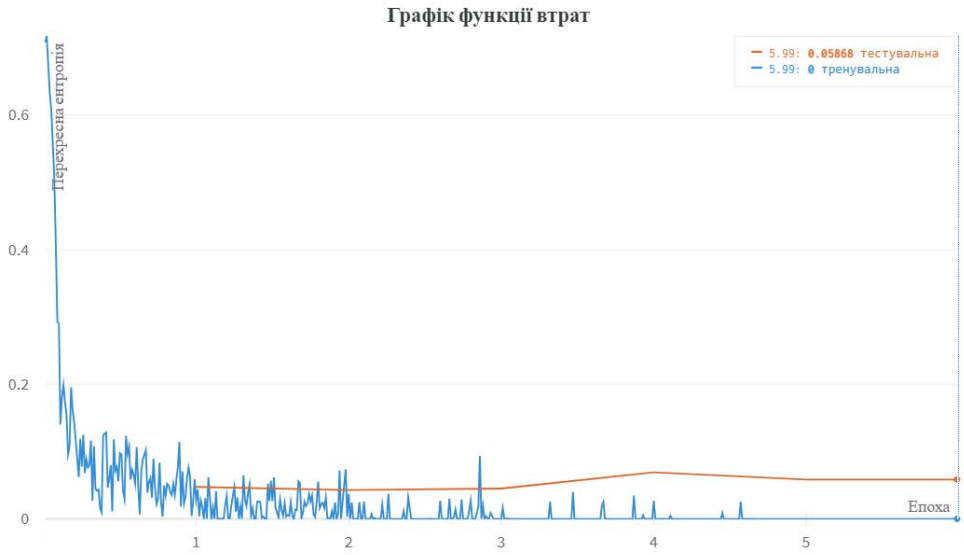


Рисунок 5.4.1.3 – Графік значення функції втрат на кожній епосі тренування моделі

Функцією втрат слугувала функція перехресної ентропії (4.4.1.4)

$$H(p,q) = -\sum_{x} p(x)\, log\ \log_{10} q(x) \qquad (4.4.1.4)$$

5.4.2 Контрольні приклади

Контрольний приклад 1

Текст:

«Для мінімізації наслідків артилерійських обстрілів попередніх років рятувальники Об’єднаних сил проводять роботи з відновлення пошкоджених житлових будинків та об’єктів інфраструктури. Упродовж доби вони відремонтували 2 будинки. Ще на 6 об’єктах роботи тривають. Українські воїни дотримуються домовленостей, досягнутих 22 липня 2020 року, в повному обсязі. Водночас наші захисники не втрачають пильності та готові миттєво реагувати на будь-які зміни оперативної обстановки.»

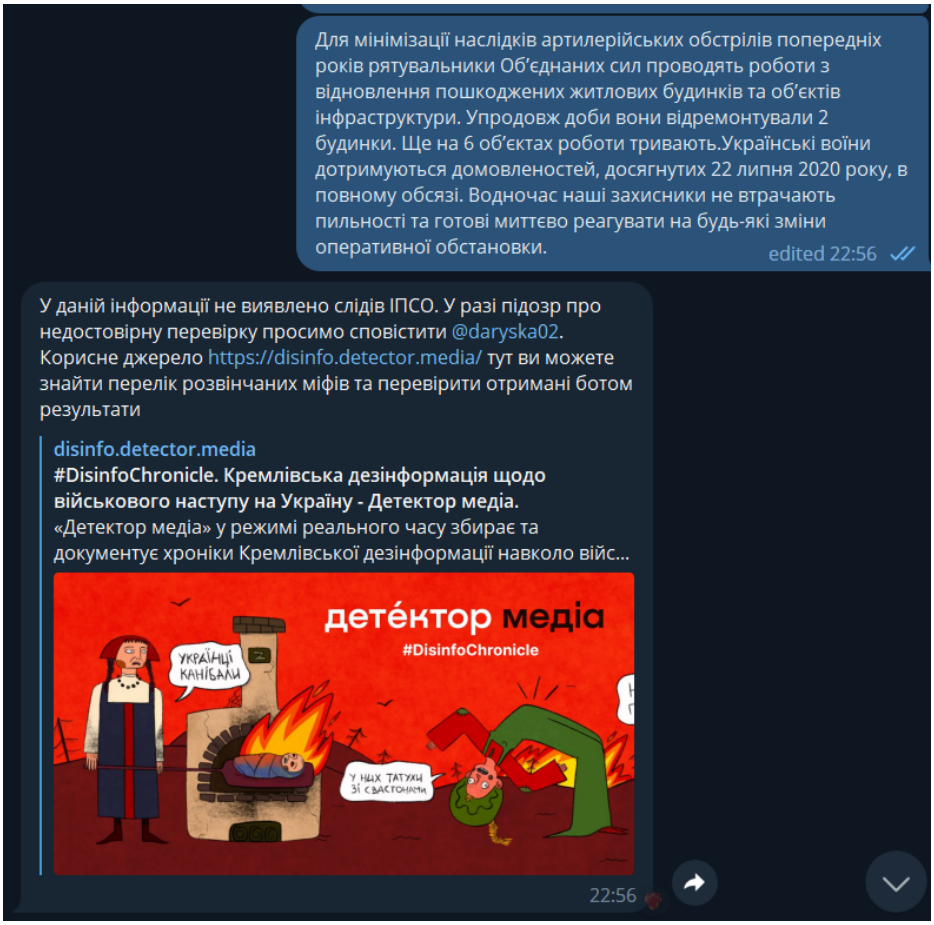


Рисунок 5.4.2.1 – Передбачення для контрольного прикладу номер 1

Подана інформація дійсно не є частиною ІПСО, адже є частиною зведення Міністерства Оборони України від 16 вересня 2020 року [25].

Контрольний приклад 2

Текст:

«Сьогодні відзначають День пам'яті юного героя-антифашиста.

Під час Великої Вітчизняної війни ці відважні хлопчики та дівчата вставали на захист Батьківщини нарівні з дорослими. Вони ставали безстрашними партизанами, підпільниками та робили небезпечні диверсії, допомагаючи вкривати поранених бійців. Вони ризикували життям і громили ворога і на суші, і у воді, і в повітрі. На весь світ вони прославили героїзм і непохитну мужність радянського народу.»

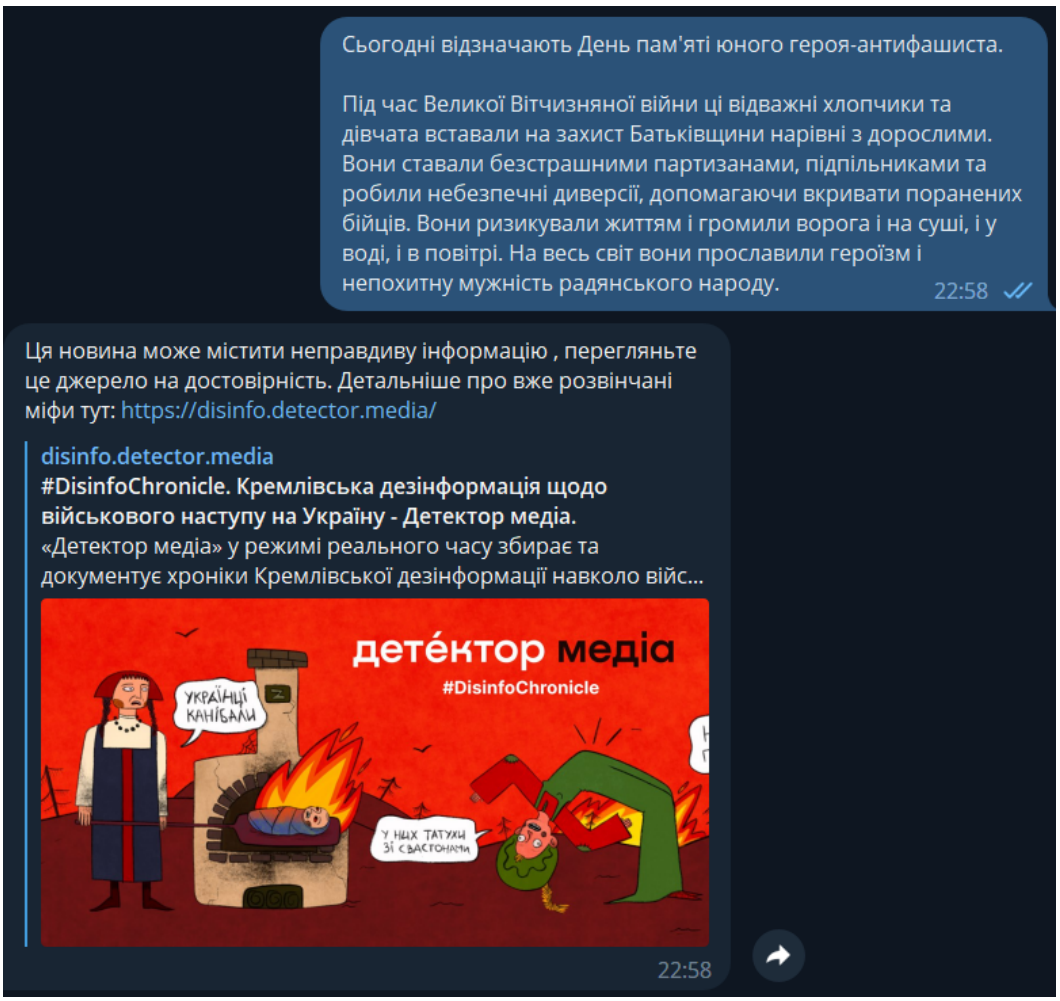


Рисунок 5.4.2.2 – Передбачення для контрольного прикладу 2

Ця новина дійсно є частиною пропагандичного посту з мережі ВКонтактє.

Контрольний приклад 3

Текст:

«Його несправедливо звинувачують у розв'язуванні війни, у смертях цивільних та загибелі військових. Але ми маємо розуміти – ця війна розв'язана не їм і не 14 днів тому. Але саме він ухвалив рішення закінчити війну, припинити цей дикий кругообіг смертей та поставити крапку у питанні статусу України. І в цьому і полягає несправедливість - лише він та ще кілька чиновників та силовиків насправді розуміють від чого вони вберегли світ цією спецопераційою.»

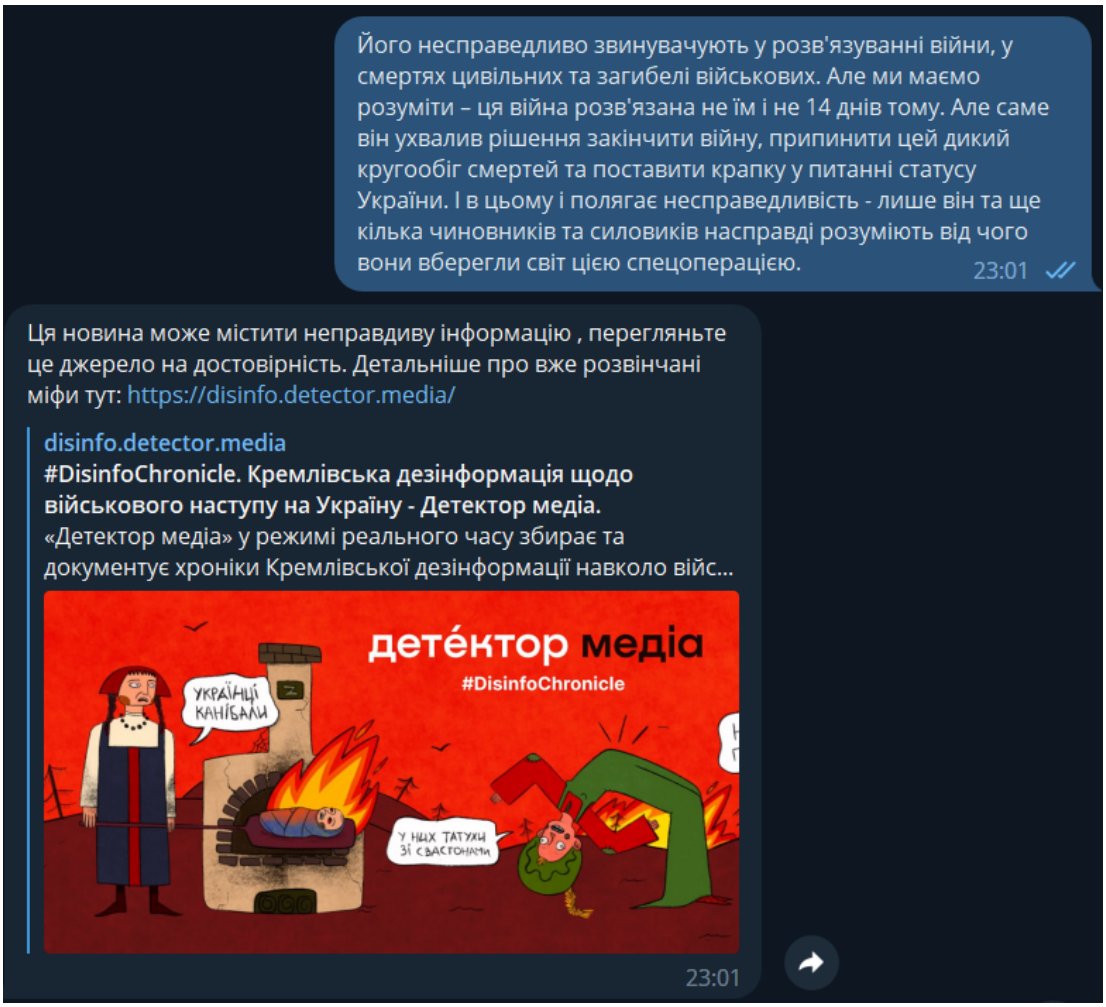


Рисунок 5.4.2.3 – Передбачення для 3 контрольного прикладу

Цей пост дійсно є перекладом тез однієї з проросійських блогерок у мережі ВКонтактє.

Контрольний приклад 4

Текст:

«окупанти обстріляли (https://hromadske.ua/posts/ukrayinsk-na-donechchini-zaznav-artobstrilu-troye-lyudej-zaginuli-zokrema-ditina) Українськ на Донеччині. Постраждали цивільні. Відомо про трьох загиблих, серед яких одна дитина. Також троє дітей поранені — ОП.

UPD: Унаслідок артобстрілу Українська поранено 4 дитини та дорослий. Їх доправили до лікарні. У місті пошкоджені 2 приватних будинки та 14 багатоповерхівок — міська військова адміністрація

hromadske | новини тут (https://t.me/hromadske_ua)»

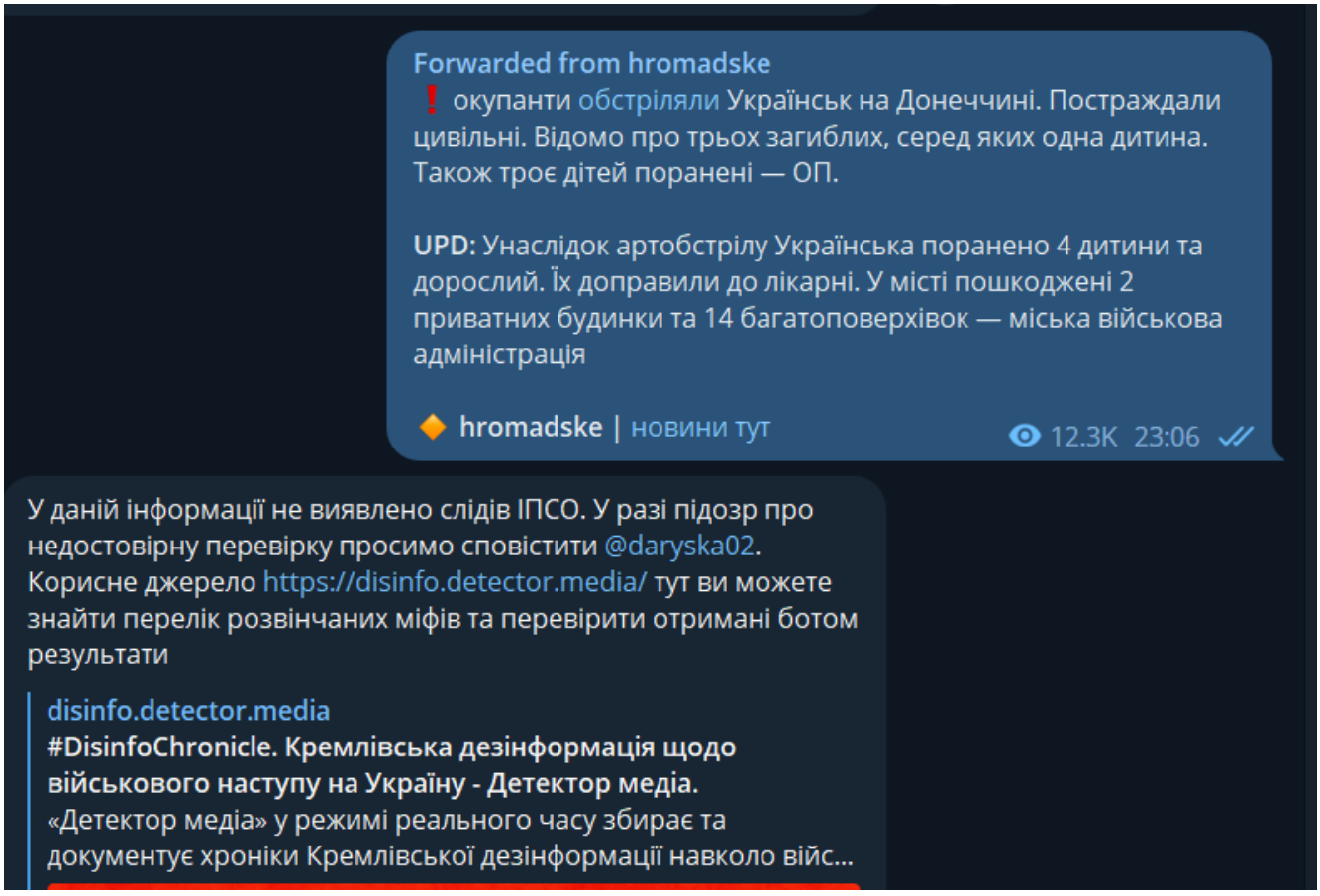


Рисунок 5.4.2.4 – Передбачення для 4го контрольного прикладу

Ця новина дійсно не є ІПСО, адже інформація є цитатою звіту Офісу Президента.

Контрольний приклад 5

Текст:

«Йому могло бути зараз 7 років. Ще сьогодні рівно 4 роки, як розіп'яли 3-річного хлопчика у трусиках у центрі Слов'янська. Це було жахливо, боляче згадувати. Все відбувалося на очах матері. Як Ісуса... А потім ще зробили надрізи. Пам'ятаємо. Сумуємо", - написав у Twitter публіцист Олександр Тверський.

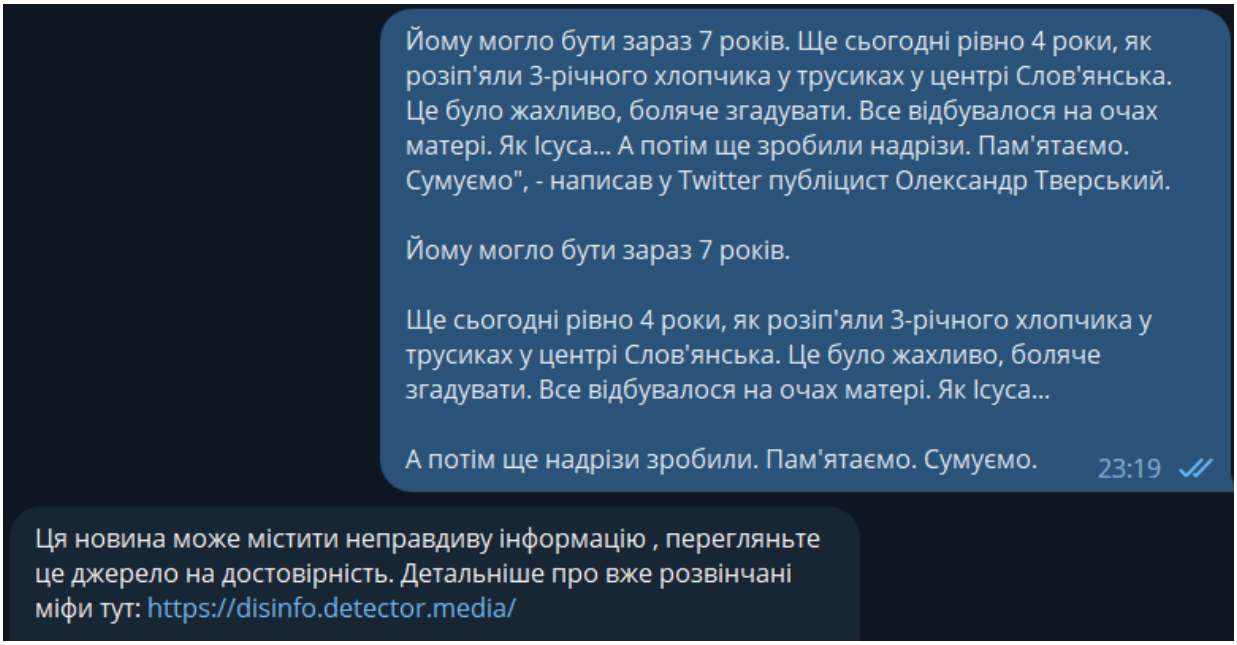


Рисунок 5.4.2.5 – Передбачення для контрольного прикладу 5

Ця відома історія про «хлопчика в трусіках» дійсно є частиною інформаційно-психологічної операції.

5.5 Висновки до розділу

В цьому розділі було розглянуто процес розробки індивідуальної системи розпізнавання ІПСО на основі переднатренованої моделі RoBERTa. Ця система має на меті виявлення та класифікацію дезінформаційних новин з високою точністю.

Було успішно інтегровано цю систему у телеграм бот, що дозволяє користувачам швидко та зручно перевіряти новини на достовірність прямо у своєму улюбленому месенджері. Користувачі можуть скористатися командами /start та /help для початку роботи з ботом та отримання додаткової інформації про систему та контакти розробника.

Розроблена система демонструє високу точність класифікації ІПСО, завдяки використанню потужної моделі RoBERTa. Вона дозволяє користувачам бути більш обізнаними та свідомими споживачами інформації, надаючи їм зручність та надійність у перевірці новинних повідомлень.

У майбутньому планується розширення функціоналу системи шляхом додавання нового тегу /information, який міститиме основні правила інформаційної гігієни та приклади дезінформаційних новин. Це дозволить ще більше покращити інформаційну грамотність користувачів та забезпечити їх захищеність від шкідливих впливів дезінформації.

Підсумовуючи, розроблена система розпізнавання ІПСО на базі RoBERTa, інтегрована у телеграм бот, є потужним інструментом, який сприяє покращенню інформаційної грамотності та захисту користувачів від дезінформації.

ВИСНОВКИ

1. Зібрано та підготовано корпус текстів, що містять ІПСО. Дані були зібрані на основі відкритих джерел у мережі інтернет. Розмір набору сягає 38 тисяч записів. Зібраний датасет було опубліковано на ресурсі спільного доступу Hugging Face.

2. Розроблено та навчено модель глибинного навчання для виявлення ІПСО у текстах. Серед низки розглянутих варіантів для реалізації було обрано модель RoBERTa. Модель було профільно натреновано на задачу класифікації тексту. Точність класифікації сягнула 99%. Тренування моделі протягом 5 епох тривало 5 годин.

3. Реалізовано програмне забезпечення, що використовує навчену модель для виявлення ІПСО.

4. Проведено тестування розробленого програмного забезпечення на різних даних для оцінки його ефективності. Модель показує хороші результати як на новіших новинах, так і на новинах 1-2 річної давнини. На тестувальній вибірці модель показала точність 99.27%. Точність передбачення для новин сьогодення є меншою, оскільки модель була натренована на даних зібраних протягом січня-квітня 2022 року.

5. Розроблено інтерфейс для взаємодії користувача з програмним забезпеченням. У ході роботи було створено Телеграм-бот, що є невибагливим, корисним та зручним у користуванні. Телеграм-бот також містить посилання на сторінку з переліком останньої викритої брехні у медіа.

ПЕРЕЛІК ПОСИЛАНЬ

1. Ukrainian News Classification | Kaggle [Електронний ресурс] // Kaggle: Your Machine Learning and Data Science Community. – Режим доступу: https://www.kaggle.com/competitions/ukrainian-news-classification/data?select=train_large.csv.

2. Bayesian Machine Learning // Encyclopedia of Machine Learning and Data Mining : Посібник. – 2-ге вид. – New York, 2017. – С. 483.М.

3. Granik and V. Mesyura, "Fake news detection using naive Bayes classifier," 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), Kyiv, UKraine, 2017, pp. 900-903

4. Deokate S. B. Fake News Detection using Support Vector Machine learning Algorithm / Suchitra B. Deokate // International Journal for Research in Applied Science and Engineering Technology. – 2019. – Т. 7, № 7. – С. 438–444.

5. Shipra S. Learn About Long Short-Term Memory (LSTM) Algorithms [Електронний ресурс] / Saxena Shipra // Analytics Vidhya. – Режим доступу: https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/. – Назва з екрана.

6. Pritika Bahad, Preeti Saxena, Raj Kamal, Fake News Detection using Bi-directional LSTM-Recurrent Neural Network, Procedia Computer Science, Volume 165, 2019, Pages 74-82, ISSN 1877-0509

7. G. Anusha, G. Praveen, D. Mounika, U. S. Krishna and R. Cristin, "Detection of Fake News using Recurrent Neural Network," 2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), Ballari, India, 2022, pp. 1-5.

8. Jamal Abdul Nasir, Osama Subhani Khan, Iraklis Varlamis, Fake news detection: A hybrid CNN-RNN based deep learning approach, International Journal of

Information Management Data Insights, Volume 1, Issue 1, 2021, 100007, ISSN 2667-0968

9. P. Kontschieder, M. Fiterau, A. Criminisi and S. R. Bulò, "Deep Neural Decision Forests," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 1467-1475

10. S. Liu, H. Tao and S. Feng, "Text Classification Research Based on Bert Model and Bayesian Network," 2019 Chinese Automation Congress (CAC), Hangzhou, China, 2019, pp. 5842-5846

11. Aljawarneh S. A., Swedat S. A. Fake News Detection Using Enhanced BERT. IEEE Transactions on Computational Social Systems. 2022. С. 1–8.

12. Hani Al-Omari, Malak Abdullah, Ola AlTiti, and Samira Shaikh. 2019. JUSTDeep at NLP4IF 2019 Task 1: Propaganda Detection using Ensemble Deep Learning Models. In Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda, pages 113–118, Hong Kong, China. Association for Computational Linguistics.

13. Dhruv Khattar, Jaipal Singh Goud, Manish Gupta, and Vasudeva Varma. 2019. MVAE: Multimodal Variational Autoencoder for Fake News Detection. In The World Wide Web Conference (WWW '19). Association for Computing Machinery, New York, NY, USA, 2915–2921.

14. Tony W. GitHub - wutonytt/Fake-News-Detection: Fake News Detection using a BERT-based Deep Learning Approach. GitHub. URL: https://github.com/wutonytt/Fake-News-Detection.

15. ISOT research lab |. Home - University of Victoria. URL: https://www.uvic.ca/engineering/ece/isot/datasets/fake-news/index.php.

16. «ПЕРЕВІРКА» – бот, який виявляє фейкові новини. Ґвара Медіа: ситуація в Харкові, соціальні зміни в Україні. URL: https://gwaramedia.com/perevirka-razom-smozhemo/#steps-link.

17. Telegram – a new era of messaging. Telegram. URL: https://telegram.org/.

18. Fake News Chrome Extension. fake-news-chrome-extension. URL: https://tlkh.github.io/fake-news-chrome-extension/.

19. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N. & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

20. Devlin, Jacob & Chang, Ming-Wei & Lee, Kenton & Toutanova, Kristina. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

21. Liu, Yinhan & Ott, Myle & Goyal, Naman & Du, Jingfei & Joshi, Mandar & Chen, Danqi & Levy, Omer & Lewis, Mike & Zettlemoyer, Luke & Stoyanov, Veselin. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach.

22. Mikolov, Tomas & Chen, Kai & Corrado, G.s & Dean, Jeffrey. (2013). Efficient Estimation of Word Representations in Vector Space. Proceedings of Workshop at ICLR. 2013.

23. Selim Reza, Marta Campos Ferreira, J.J.M. Machado, João Manuel R.S. Tavares, A multi-head attention-based transformer model for traffic flow forecasting with a comparative analysis to recurrent neural networks, Expert Systems with Applications, Volume 202, 2022, 117275, ISSN 0957-4174.

24. Ustyianovych T. Propaganda and fake news on the war in Ukraine [Electronic resource] / Taras Ustyianovych, Solomiia Fedushko // Lviv Polytechnic National University. – Mode of access: https://doi.org/10.21227/z0j6-bw17

25. Вечірнє зведення щодо ситуації в районі проведення операції Об’єднаних сил станом на 17.00 16 вересня 2020 року [Електронний ресурс] // Міністерство Оборони України. – Режим доступу: https://www.mil.gov.ua/special/news.html?article=59707.

Додаток А

Лістинги програм

А.1 Лістинг користувацького інтерфейсу

```
from transformers import RobertaTokenizerFast, RobertaForSequenceClassification,Trainer, TrainingArguments
import datasets
import torch

import Constants as keys
from telegram.ext import *
from telegram import Update
from datetime import datetime
import re

model = RobertaForSequenceClassification.from_pretrained('C:/Users/dnedi/PycharmProjects/Test/News detection')
tokenizer = RobertaTokenizerFast.from_pretrained('youscan/ukr-roberta-base', max_length=520)
def tokenization(batched_text):
  return tokenizer(batched_text ['text'], padding = True, truncation=True)


welcoming_text = '''
Привіт.
Я перевіряю новини на правдивість та дивлюся, чи не є вони ворожим засобом маніпуляції.
```

```
Приємно познайомитися!
'''

text_inf = '''
Для того щоб скористатися ботом - просто перекинь сюди новину!
'''

## Responses
def handle_responses(text: str) -> str:
  text: str = text.lower()

  if text in ['привіт','прив','хай']:
    return welcoming_text

  if len(text.split())>=10:
    print('recognition')
    dict = {
      'text': [text],
      'label': [0]
    }
    inp = datasets.Dataset.from_dict(dict)
    inp = inp.map(tokenization, batched=True, batch_size=len(inp))
    inp.set_format('torch', columns= ['input_ids', 'attention_mask', 'label'])

    res = model(inp ['input_ids']).logits.argmax()
    if int(res)==0:
      return 'У даній інформації не виявлено слідів ІПСО. У разі підозр про недостовірну перевірку просимо сповістити @daryska02. Корисне джерело
```

https://disinfo.detector.media/ тут ви можете знайти перелік розвінчаних міфів та перевірити отримані ботом результати'

else:

return 'Ця новина може містити неправдиву інформацію , перегляньте це джерело на достовірність. Детальніше про вже розвінчані міфи тут: https://disinfo.detector.media/'

return 'Абракадабра зімзалабім! А тепер давай щось зрозуміле.'

Message

async def handle_message(update: Update, context: ContextTypes.DEFAULT_TYPE):

text: str = update.message.text.lower()

print(f'User {update.message.chat.first_name} has sent {text}')

response: str = handle_responses(text)

print('Bot: ', response)

await update.message.reply_text(response)

print('Bot started...')

Commands

```
async def start_command(update: Update, context: ContextTypes.DEFAULT_TYPE):
    await update.message.reply_text(welcoming_text)


async def help_command(update: Update, context: ContextTypes.DEFAULT_TYPE):
    await update.message.reply_text('If you need help, ping me @daryska02')

#async def custom_command(update: Update, context: ContextTypes.DEFAULT_TYPE):
#  await update.message.reply_text('If you need help, ping me @daryska02')

def error_log( update: Update, context: ContextTypes.DEFAULT_TYPE):
  print(f'Update {update} caused error {context.error}')


if __name__ == '__main__':
  print('Starting bot...')
  app = Application.builder().token(keys.API_KEY).build()

  #Commands
  app.add_handler(CommandHandler('start', start_command))
  app.add_handler(CommandHandler('help', help_command))
#  app.add_handler(CommandHandler('custom', custom_command))

  # Messages
  app.add_handler(MessageHandler(filters.TEXT, handle_message))
```

```
#Errors
app.add_error_handler(error_log)
print('Polling...')
app.run_polling(poll_interval=20)
```

А.2 Лістинг підготовки даних

```
"#!/usr/bin/env python
# coding: utf-8

from datasets import load_dataset

dataset = load_dataset('zeusfsx/ukrainian-news')

dataset ['train'] [0:1]

dataset.column_names

dataset.num_rows

pd.DataFrame(dataset.unique('owner')).head(10)

info = ['Офіс Президента України', 'Zelenskiy / Official', 'МВС України', 'Суспільне', 'Громадське', 'ДСНС', 'Ліга', 'НВ', 'Генеральний штаб Збройних сил України', 'Кабінет Міністрів України', 'Офіс Президента України', 'Територіальна оборона ЗСУ', 'Міністерство інфраструктури України', 'Укрзалізниця', 'Міністерство оборони України', 'Міністерство внутрішніх справ', 'Національна поліція
```

України','Державна служба надзвичайних ситуацій (ДСНС)', 'Державна прикордонна служба України', 'Центр стратегічних комунікацій та інформаційної безпеки', "Державна служба спеціального зв'язку та захисту інформації України", 'Сухопутні війська ЗС України', 'Військово-морські сили ЗСУ', 'Укравторух','Укравтодор', 'Державна служба України з безпеки на транспорті (ДСБТ)']

```
import pyarrow as pa
import pyarrow.compute as comp

table = dataset ['train'].data

flags = comp.is_in(table ["owner"], value_set = pa.array(info))
filter_table = table.filter(flags)
filter_table.to_pandas().head(10)

filter_table.to_pandas().to_csv('NOT PSYCOPS.csv')

import pandas as pd

data = pd.read_csv('information-war.csv')
## 01.02.2022 until 26.04.2022.
data.head()

data.info()


data.shape
```

```
data_no = pd.read_csv('NOT PSYCOPS.csv')

data_no.info()

labeled_data = pd.DataFrame()

labeled_data = pd.concat( [labeled_data,data_no ['text']])

labeled_data ['is_psyops'] = 0

labeled_data.columns = ['text','is_psyops']

shuffled_ipso = data.sample(frac=1)

shuffled_cut_text = pd.DataFrame(shuffled_ipso [shuffled_ipso ['post_type']=='post'] ['text'].head(data_no.shape [0]))


shuffled_cut_text.shape


shuffled_cut_text ['is_psyops'] = 1


shuffled_cut_text

from googletrans import Translator, constants

translator = Translator()
```

```
trans = translator.translate('Привет',dest = 'uk', src='ru')

shuffled_cut_text ['text_ukr'] = shuffled_cut_text ['text'].apply(translator.translate, dest = 'uk', src='ru').apply(getattr, args=('text',))

shuffled_cut_text.head(20)

shuffled_cut_text.to_csv('translates ipso.csv')



shuffled_cut_text.columns= ['text_rus', 'is_psyops', 'text']

labeled_data = pd.concat( [labeled_data,shuffled_cut_text [ ['text','is_psyops']]])

labeled_data.info()



import re

labeled_data ['text'] = labeled_data ['text'].apply(lambda st: re.sub("<.*?>|\ [.*?]","",st))
labeled_data ['text'] = labeled_data ['text'].apply(lambda st: re.sub("\\n"," ",st))
labeled_data ['text'] = labeled_data ['text'].apply(lambda st: re.sub("#Repost\s|@ [a-zA-Z]+\s","",st))

labeled_data [labeled_data ['is_psyops']==1].iloc [6].text
```

```
labeled_data.to_csv('data_labeled.csv')

labeled_data.info()

import pandas as pd
data_l = pd.read_csv('data_labeled.csv')


data_l = data_l.dropna()


from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split( data_l ['text'], data_l ['is_psyops'], test_size=0.33, random_state=42)


train_data = pd.concat( [X_train, y_train], axis=1)
test_data = pd.concat( [X_test, y_test], axis=1)
cols = ['text', 'label']
train_data.columns = cols
test_data.columns = cols


train_data

train_data.to_json('train_data.jsonl', orient='records', lines=True, force_ascii=False)
test_data.to_json('test_data.jsonl', orient='records', lines=True, force_ascii=False)
```

А.3 Лістинг донавчання моделі

```
#%%
import pandas as pd
import datasets
from datasets import Dataset, DatasetDict
from transformers import RobertaTokenizerFast, RobertaForSequenceClassification,Trainer, TrainingArguments
import torch.nn as nn
import torch
from torch.utils.data import Dataset, DataLoader
import numpy as np
from sklearn.metrics import accuracy_score, precision_recall_fscore_support

import os
import wandb
run = wandb.init()
#%%
train_data, test_data = datasets.load_dataset('Meduzka/ukr_psyops_false_news', split = ['train', 'test'])
#%%
datasets.__version__
#%%
#from transformers import AutoTokenizer, AutoModelForMaskedLM

#tokenizer = AutoTokenizer.from_pretrained("youscan/ukr-roberta-base")

#model = AutoModelForMaskedLM.from_pretrained("youscan/ukr-roberta-base")
#%%
```

```
model = RobertaForSequenceClassification.from_pretrained('youscan/ukr-roberta-base')

tokenizer = RobertaTokenizerFast.from_pretrained('youscan/ukr-roberta-base', max_length = 512)
#%%
# define a function that will tokenize the model, and will return the relevant inputs for the model
def tokenization(batched_text):
  return tokenizer(batched_text ['text'], padding = True, truncation=True)


train_data = train_data.map(tokenization, batched = True, batch_size = len(train_data))
test_data = test_data.map(tokenization, batched = True, batch_size = len(test_data))
#%%
train_data.set_format('torch', columns= ['input_ids', 'attention_mask', 'label'])
test_data.set_format('torch', columns= ['input_ids', 'attention_mask', 'label'])
#%%
def compute_metrics(pred):
  labels = pred.label_ids
  preds = pred.predictions.argmax(-1)
  precision, recall, f1, _ = precision_recall_fscore_support(labels, preds, average='binary')
  acc = accuracy_score(labels, preds)
  return {
    'accuracy': acc,
    'f1': f1,
    'precision': precision,
```

```
    'recall': recall
  }
#%%
# define the training arguments
training_args = TrainingArguments(
  output_dir = '/results',
  num_train_epochs=8,
  per_device_train_batch_size = 2,
  gradient_accumulation_steps = 16,
  per_device_eval_batch_size= 8,
  evaluation_strategy = "epoch",
  save_strategy = 'epoch',
  disable_tqdm = False,
  load_best_model_at_end=True,
  warmup_steps=500,
  weight_decay=0.01,
  logging_steps = 8,
 #  fp16 = True,
  logging_dir='/logs',
  dataloader_num_workers = 8,
  run_name = 'roberta-classification',
  optim="adamw_torch"
)

#%%
# instantiate the trainer class and check for available devices
trainer = Trainer(
  model=model,
  args=training_args,
```

```
  compute_metrics=compute_metrics,
  train_dataset=train_data,
  eval_dataset=test_data
)
device = 'cuda' if torch.cuda.is_available() else 'cpu'
device
#%%
# train the model

torch.cuda.empty_cache()
np.object = object
trainer.train()

#%%
trainer.evaluate()
run.finish()
#%%
torch.save(model.state_dict(), 'RoBERTa on news.pt')
#%%
y_pred = model()
#%%
trainer.save_model('new news class')
#%%



#%%
def tokenization(batched_text):
  return tokenizer(batched_text ['text'], padding = True, truncation=True)
#%%
```

text_inp = '"Окупанти почали реалізовувати провокацію із застосуванням хімічної зброї на тимчасово окупованих територіях Запорізької області — ГУР. (https://t.me/DIUkraine/2357) "Реальними жертвами ворожої провокації стануть солдати російської окупаційної армії. Сліди ураження хімічною зброєю на їхніх тілах держава-агресор планує використати як фейковий доказ для звинувачення України", — пише розвідка"'

text_inp = '"Вища рада правосуддя надала дозвіл на арешт судді Олексія Тандира, який насмерть збив нацгвардійця.

Під час засідання Тандира заявив, що не був п'яний і не вживав напередодні, а походження пляшки в авто йому невідоме.

Також відомо, що суддя надав лише частину біологічних зразків"'

text_inp = '"Глава ГУР України заявив про намір знищити 3 млн. жителів Криму.

-Після перемоги поїду до Севастополя, це моє рідне місто. Роботи буде дуже багато. Ми отримали 3 млн. людей, які жили під пропагандою РФ. Це видозмінені люди, на яких чекає фізичне знищення за їхні вчинки», - сказав він.

Ви тепер розумієте, чому ми так у Криму реагуємо на ждунів і вимагаємо їх покарання?

Це посібники нацистів, які бажають нам смерті, або вони або ми іншого не дано!

Привезіть, будь ласка, цього бандерівця до Севастополя і жителі розірвуть його на площі Нахімова."'

text_inp = '"1992-го утворився «Промінь», із редакторів музичної редакції: значна частина пішла на «Промінь», інша залишилася на першому каналі — фактично розділилися порівну. На першому каналі музична редакція теж мала свої відрізки, а «Промінь» був цілком музичним, туди пішли хороші редактори з

музичної редакції: Алік Горський, Микола Амосов, Вероніка Маковій, Саша Васильєв, Іра Бондаренко — люди, які виросли в музичній редакції. Всі вони були з вищою музичною освітою, підхопили ідею втілювати все за принципом польського радіо. Микола Амосов розписав сітку! Пам’ятаю, як це було: хітпарад уранці, потім були інформаційно-музичні відрізки. Вважалося, що з 11 по 12-ту годину слухають домогосподарки, тому там було ретро. В обідній час — програма «На ваші замовлення»: отримували листи, заявки, їх озвучували й давали пісні. Далі йшла закордонна музика, о 18-й був Коган із годиною меломана, потім джаз Діми Гальона... Багато було людей, яких привели музичні редактори як своїх друзів. Тоді розмилося поняття, що редактор має бути людиною з профільною вищою музичною освітою: прийшли меломани, аматори, які мали колекції дисків. Розмилося й поняття, що в ефірі має звучати тільки те, що є у фонотеці та прийнято художньою радою. На офіційному рівні почали працювати зі студією «Level», звучав хітпарад цієї студії, почали працювати з фірмами звукозапису, які тоді почали виникати.'"

```
dict = {
  'text': [text_inp],
  'label': [0]
}

inp = datasets.Dataset.from_dict(dict)

inp = inp.map(tokenization, batched = True, batch_size = len(inp))
inp.set_format('torch', columns= ['input_ids', 'attention_mask', 'label'])
#%%
int(model(inp ['input_ids']).logits.argmax())

#%%
```

```
#%%
inp ['input_ids']
#%%
```

Додаток Б

Ілюстративний матеріал

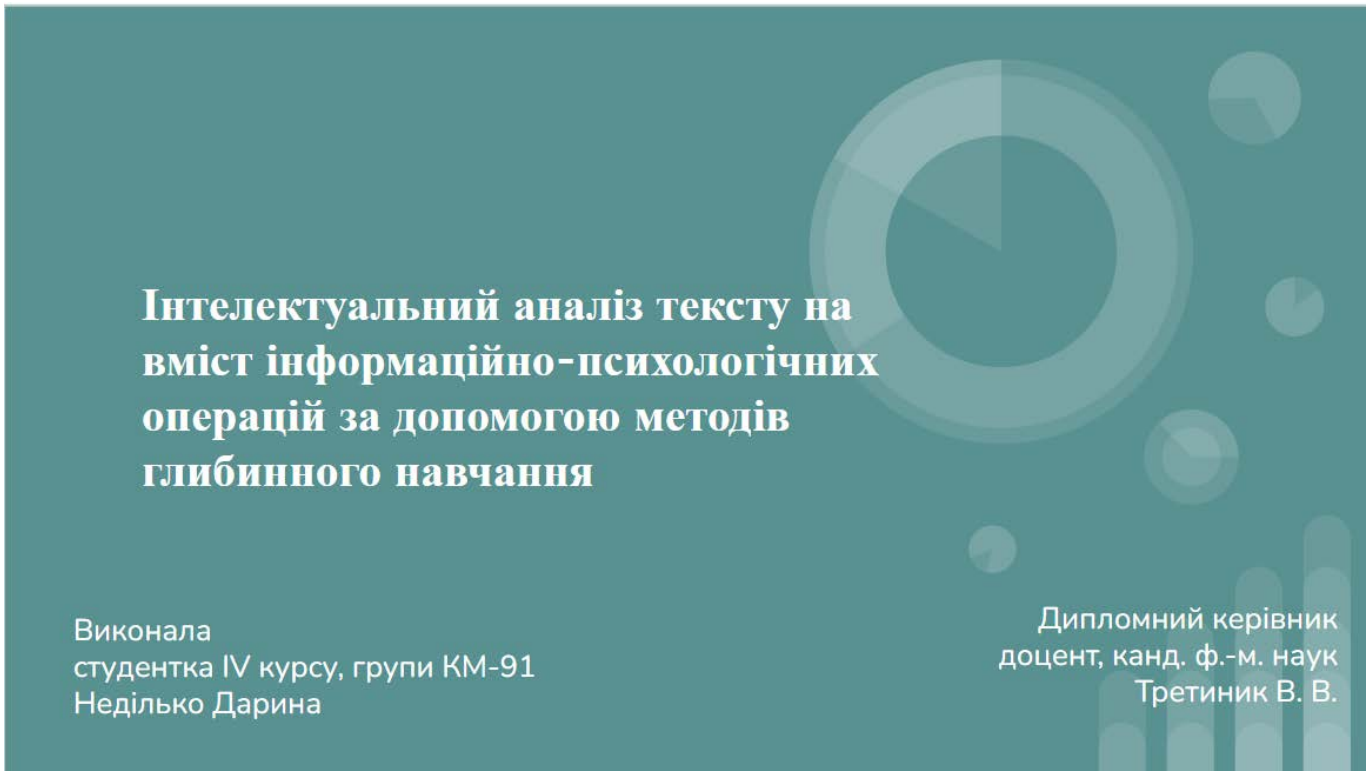


Рисунок Б.1 – Слайд 1

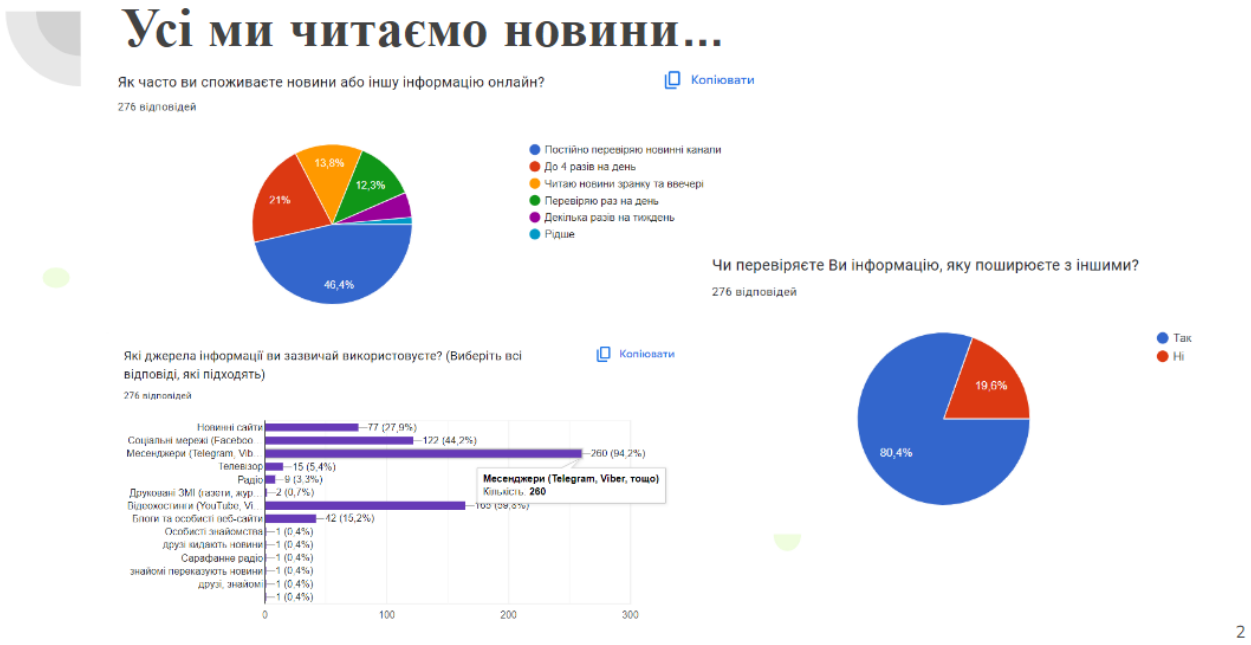


Рисунок Б.2 – Слайд 2

Рисунок Б.3 – Слайд 3

Огляд комерційних рішень

Ресурс	К1	К2	К3	К4	К5	К6
“Перевірка”	Зручно	Так	Легко	Так	Так	Точно, не швидко
“Fake News Detection”	Лише на ПК у браузері	Ні	Вимагає додаткових знань	Так, але американоцентричний погляд	Так	Швидко та точно
Наявна розробка	Зручно	Так	Легко	Так	Так	Швидко та точно

К1 - Зручність використання, К2 - наявність українського інтерфейсу, К3 - легкість встановлення, К4 - перевірка українських новин, К5 - наявність інструкцій до використання, К6 - точність та швидкість перевірки

4

Рисунок Б.4 – Слайд 4

Огляд програмних рішень

Модель[1]	Токенізатор	Точність	Precision	Recall	F1 Score
CNN	BERT	0.8993	0.9342	0.8875	0.9103
BiLSTM	BERT	0.9065	0.9589	0.8750	0.9193
BERT	BERT	0.9137	0.9359	0.9125	0.9241
RoBERTa	RoBERTa	0.9209	0.9259	0.9375	0.9317
		RoBERTa	BiLSTM	RoBERTa	RoBERTa

Precision = True Positives/Predicted Positives
Recall = True Positive/Real Positives
F1 Score = 2*Precision*Recall/(Precision+Recall)

5

Рисунок Б.5 – Слайд 5

Рисунок Б.6 – Слайд 6

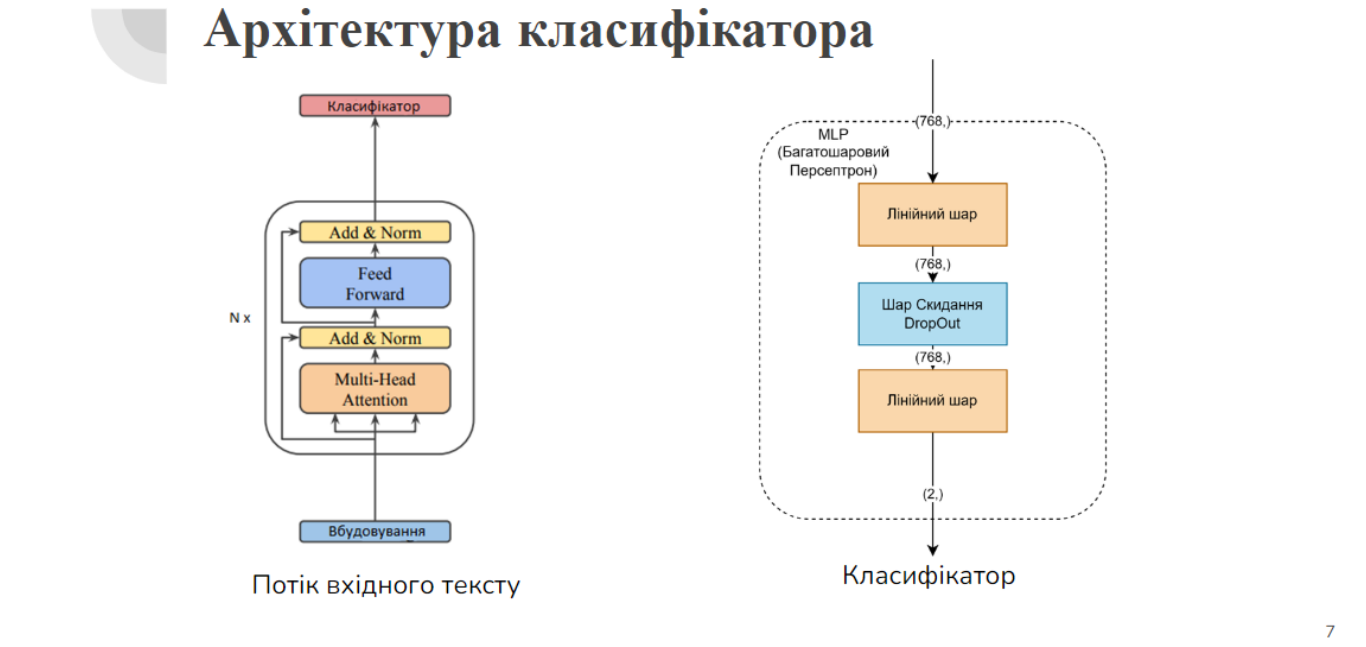
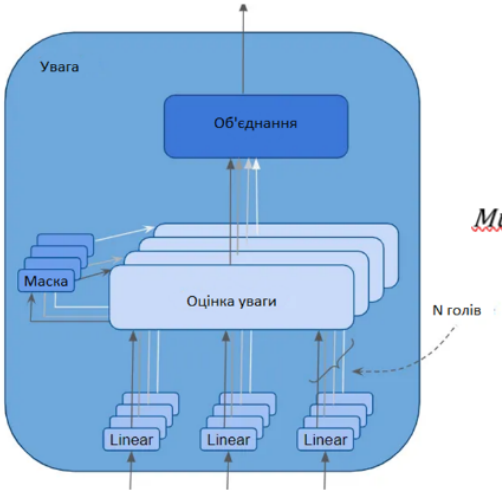


Рисунок Б.7 – Слайд 7

Механізм уваги



Багатоголівна увага

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

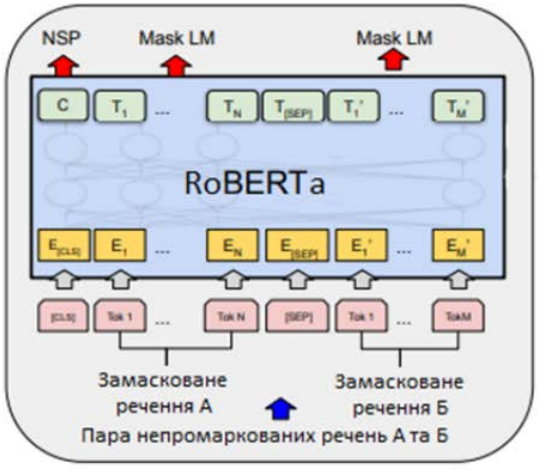
$$MultiHead(Q,K,V) = Concat(head_{1..h})W^O \text{ (3.2.2.3), де}$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \text{ (3.2.2.4)[22]}$$

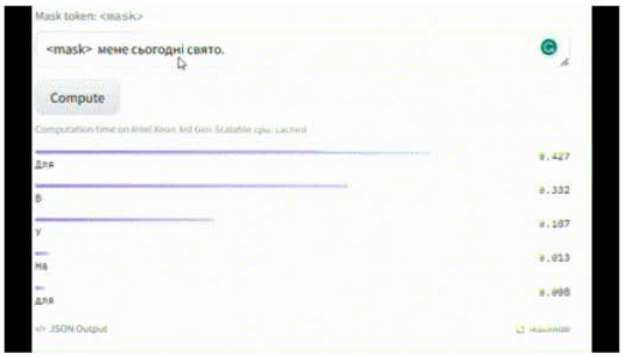
8

Рисунок Б.8 – Слайд 8

RoBERTa - Попереднє тренування



Попереднє тренування



Приклад роботи задачі Маскування мови

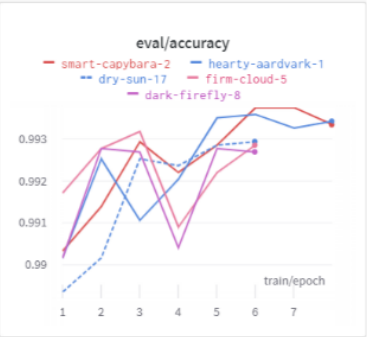
9

Рисунок Б.9 – Слайд 9

Профільне тренування

```
train_data, test_data = datasets.load_dataset('Meduzka/ukr_psyops_false_news', split =['train', 'test'])
```

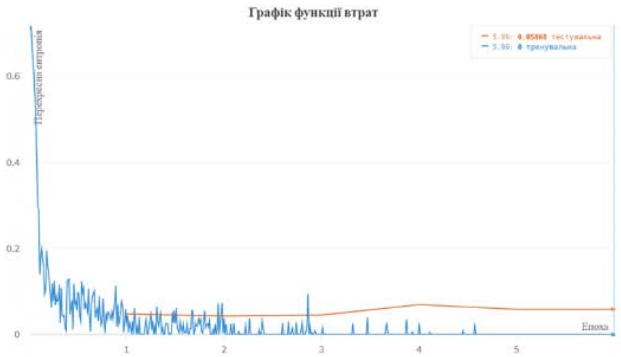
Validation Loss	Accuracy	F1	Precision	Recall
0.047803	0.990179	0.990222	0.985048	0.995451
0.043149	0.992777	0.992772	0.992530	0.993014
0.045322	0.992695	0.992686	0.993009	0.992364
0.069402	0.990423	0.990366	0.995405	0.985378
0.058773	0.992777	0.992779	0.991572	0.993989
0.058675	0.992695	0.992682	0.993651	0.991714



10

Рисунок Б.10 – Слайд 10

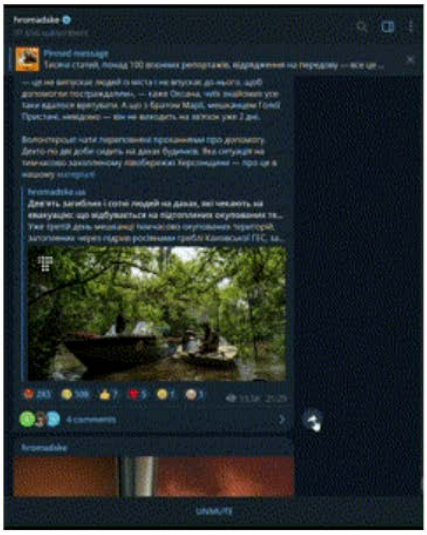
Показники моєї моделі



11

Рисунок Б.11 – Слайд 11

Приклад взаємодії



12

Рисунок Б.12 – Слайд 12

Висновки

- Зроблено дослідження потреб суспільства у інструменті перевірки інформації
- Опрацьовано методи розпізнавання ІПСО
- Підготовлено датасет з правдивими та фальшивими новинами
- Натреновано модель RoBERTa для задачі класифікації тексту, точність передбачення 99%
- Створено інтерфейс у вигляді чат-бота у Телеграм

13

Рисунок Б.13 – Слайд 13

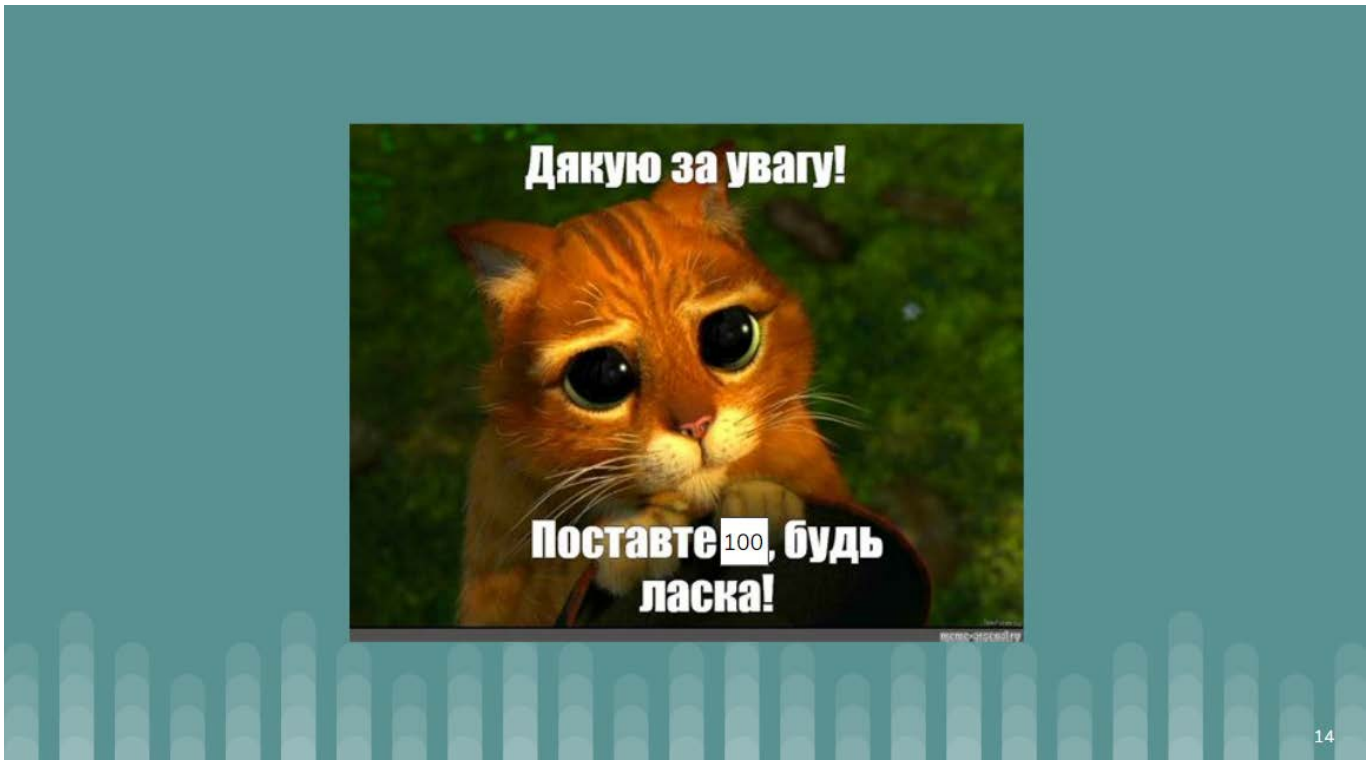


Рисунок Б.14 – Слайд 14

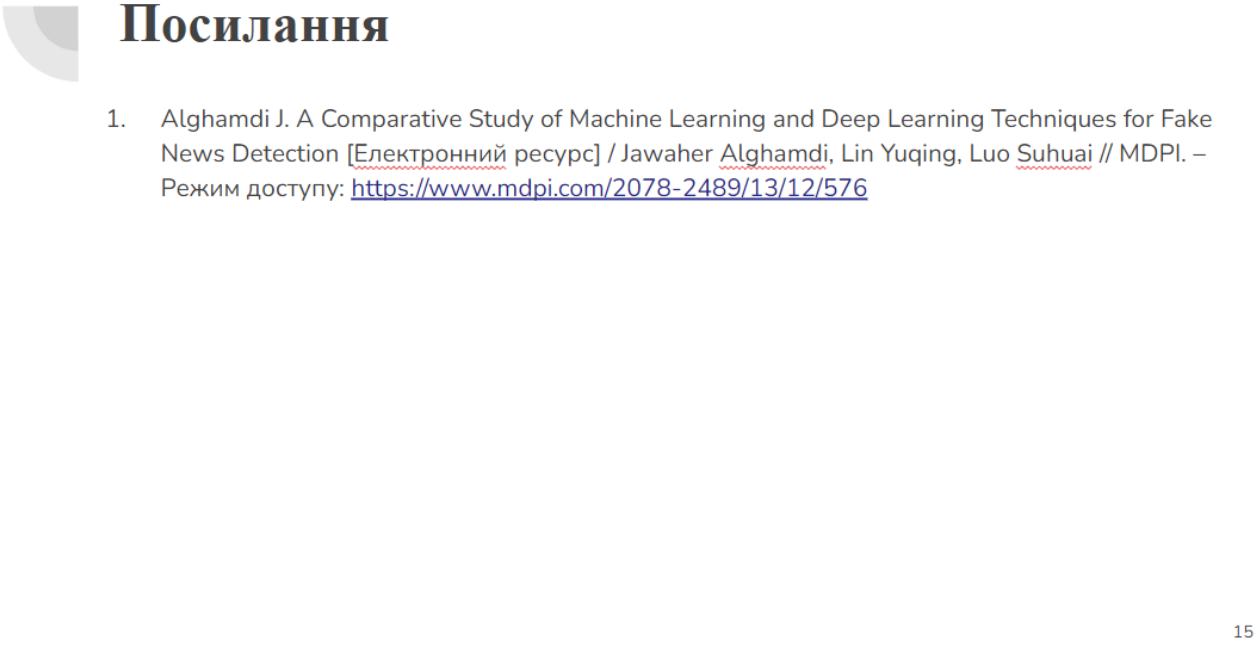


Рисунок Б.15 – Слайд 15

Додаток А - Приклад токенізованого тексту

```
Я люблю токенізувати текст
tensor([    0,   848, 12642, 32422,   296,   456,  1256,  6849,     2])
tensor([1, 1, 1, 1, 1, 1, 1, 1, 1])
Реконструкція:
 ['<s>', 'Ð¯', 'ĠÐ»ÑİÐ±Ð»Ñİ', 'ĠÑĤÐ¾Ð°', 'ÐµÐ¾', 'ÑĸÐ·', 'ÑĥÐ²Ð°ÑĤÐ¸', 'ĠÑĤÐµÐºÑģÑĤ', '</s>']
 <s>Я люблю токенізувати текст</s>


Моя улюблена страва <mask>
tensor([    0, 19848, 42655, 30052,     4,     2,     1,     1,     1])
tensor([1, 1, 1, 1, 1, 1, 0, 0, 0])
Реконструкція:
 ['<s>', 'ÐlÐ¾Ñı', 'ĠÑĥÐ»ÑİÐ±Ð»ÐµÐ¾Ð°', 'ĠÑģÑĤÑĢÐ°Ð²Ð°', '<mask>', '</s>', '<pad>', '<pad>', '<pad>']
 <s>Моя улюблена страва<mask></s><pad><pad><pad>
```

16

Рисунок Б.16 – Слайд 16