НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

«На правах рукопису»
УДК _004.8_

До захисту допущено:
Завідувач кафедри
_____ Олександр РОЛІК
«___»_____2024 р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою
«Інформаційне забезпечення робототехнічних систем»

зі спеціальності 126 «Інформаційні системи та технології»

на тему: «Інтелектуальний робот-асистент на основі великих мовних моделей для виявлення пропаганди»

Виконала:
студентка 2 курсу, групи ІК-32мп
Захарчин Надія Романівна _____

Керівник:
доцент каф. ІСТ, к.т.н., доц.
Олійник Володимир Валентинович _____

Рецензент:
доцент каф. ІПІ, к.т.н., доц.
Лісовиченко Олег Іванович _____

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

Київ – 2024 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційне забезпечення робототехнічних систем»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Олександр РОЛІК
«___»_____2024 р.

ЗАВДАННЯ
на магістерську дисертацію студентці
Захарчин Надії Романівні

1. Тема дисертації «Інтелектуальний робот-асистент на основі великих мовних моделей для виявлення пропаганди», науковий керівник дисертації Олійник Володимир Валентинович, к.т.н., доц., затверджені наказом по університету від «08» 11 2024 р. № 5016-с

2. Термін подання студентом дисертації «09» 12 2024 р.

3. Об’єкт дослідження: Виявлення пропаганди у текстах англійською та російською мовами.

4. Вихідні дані: надійність від 95%, доступність від 95%, сумісність інтелектуального компонента з хардвер-пристроями.

5. Перелік завдань, які потрібно розробити: провести аналіз існуючих рішень, обрати великі мовні моделі для експериментів, провести експериментальне дослідження, реалізувати інтелектуального асистента з використанням найкращої в ході експериментів моделі, підготувати текстову та графічну частину пояснювальної записки.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: діаграма варіантів використання, схема архітектури трансформера, схема архітектури моделей Gemma, інструкції для точного налаштування, схема доналаштованої моделі Gemma з використанням адаптера, схема інтелектуального асистента, приклади роботи інтелектуального асистента, метрики.

7. Орієнтовний перелік публікацій:

8. Дата видачі завдання 02.09.2024 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз попередніх досліджень, методів і підходів в предметній області	01.07	
2	Постановка конкретної задачі	30.07	
3	Пошук та збір набору даних	20.08	
4	Дослідження ефективності першого набору моделей	20.09	
5	Дослідження ефективності другого набору моделей	10.10	
6	Порівняльний аналіз моделей	25.10	
7	Розробка інтелектуального асистента	10.11	
8	Оформлення документації	20.11	
8	Подання роботи на попередній захист	25.11	
9	Подання роботи на основний захист	09.12	

Студент (-ка) Надія ЗАХАРЧИН

Науковий керівник Володимир ОЛІЙНИК

РЕФЕРАТ

Інтелектуальний робот-асистент на основі великих мовних моделей для виявлення пропаганди: 103 с., 28 табл., 58 рис., 9 дод., 37 джерел.

ВИЯВЛЕННЯ ПРОПАГАНДИ, ВЕЛИКІ МОВНІ МОДЕЛІ, МЕДІАГРАМОТНІСТЬ, ОСВІТНІЙ ПОМІЧНИК, ТЕХНІКИ ПРОПАГАНДИ, ІНФОРМАЦІЙНА ВІЙНА, ПРОПАГАНДА В НОВИНАХ.

В час інтенсивного впливу інформаційних ресурсів на свідомість людини надзвичайно є важливим розвиток медіаграмотності та навичок критичного мислення. Сюди входить і таке завдання, як виявлення пропаганди у текстовому контенті (наприклад, новинах, дописах у соцмережах), для якого можна і доцільно досліджувати можливість автоматизації з використанням технологій штучного інтелекту. Потенціал вирішення цієї задачі показують великі мовні моделі, найбільші з яких виникли відносно нещодавно і ще не були застосовані до неї.

Метою дослідження є розробка інтелектуального компонента робота з виявлення пропаганди на основі великої мовної моделі для використання в освітньому процесі.

Об’єктом дослідження є виявлення пропаганди у текстах англійською та російською мовами.

Предметом дослідження є застосування великих мовних моделей для виявлення пропаганди.

Методами дослідження є аналіз наявних праць, експерименти та порівняння над великими мовними моделями для завдання виявлення пропаганди.

Науковою новизною одержаних результатів є застосування двомовного набору даних для розробки інтелектуального асистента, експерименти і порівняння над такими великими мовними моделями, як GPT-4o mini і моделі сімейства Gemma/Gemma 2.

Результати роботи можна використовувати в інтеграції з роботом у освітньому процесі для навчання школярів медіаграмотності та характеристикам технік пропаганди.

ABSTRACT

Intelligent assistant robot based on large language models for propaganda detection : 103 p., 28 tab., 58 draw., 9 app., 37 sources.

PROPAGANDA DETECTION, LARGE LANGUAGE MODELS, MEDIA LITERACY, EDUCATIONAL ASSISTANT, PROPAGANDA TECHNIQUES, INFORMATION WARFARE, PROPAGANDA IN NEWS.

At a time of intense influence of information resources on human consciousness, the development of media literacy and critical thinking skills is extremely important. This includes such a task as detecting propaganda in the news, for which it is reasonable to explore the possibility of automation using artificial intelligence technologies. As this task involves working with text, the potential for solving this task is shown by large language models, the largest of which have emerged relatively recently and have not yet been applied to propaganda detection.

The aim of the study is to develop an intelligent assistant for propaganda detection based on a large language model for use in the educational process with the possibility of integrating it with a robot.

The object of the study is the detection of propaganda in news texts in English and Russian.

The subject of the study is the application of large-scale language models for propaganda detection.

The research methods are an analysis of existing works, experiments and comparisons on large-scale language models for the task of propaganda detection.

The scientific novelty of the results is the use of a bilingual dataset for the development of an intelligent assistant, experiments and comparisons on such large language models as GPT-4o mini and Gemma/Gemma 2 family models.

The results of the work, in particular the intelligent assistant, can be used in the educational process to teach students media literacy and the characteristics of propaganda techniques.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

LLM – велика мовна модель, великі мовні моделі (Large Language Models);

GPT – генеративний передтренований трансформер (Generative Pre-trained Transformer);

NLP – обробка природної мови (Natural Language Processing);

ШІ – штучний інтелект.

ВСТУП

Термін «інформаційна війна» в останнє десятиліття сформувався і набув значного поширення внаслідок таких чинників, як процеси глобалізації і стан технологій. Він не втрачає актуальності в світлі воєн та конфліктів у світі та усвідомлення глобальними політичними силами значимості та впливу інформаційних технологій на людство.

Пропаганда – форма комунікації, спрямована на поширення в суспільстві – світогляду, теорії, твердження, фактів, аргументів, чуток та інших відомостей для впливу на суспільну думку на користь певної спільної справи чи громадської позиції [1]. Пропаганда зустрічається у різних формах і в багатьох джерелах, до яких належать, наприклад, новинні сайти і соцмережі.

Будучи у постійному потоці інформації, дуже легко піддаватися пропаганді, не усвідомлюючи цього. Такі фактори, як вік та освіта можуть впливати на вразливість конкретної людини до сприйняття пропаганди, довіри до фейків та поширювання дезінформації. До потенційно вразливих верств, зокрема, належать підлітки, через їх постійну залученість до інформаційного простору через соцмережі.

В світлі сучасних подій в Україні та світі, а також російсько-української війни, яка містить також інформаційне поле, важливим завданням є не тільки ідентифікувати пропаганду, але й просувати у суспільстві медіаграмотність та виховувати критичне мислення. Це доцільно впроваджувати у освітніх закладах, зокрема, школах.

Для забезпечення більшої зацікавленості та залученості учнів у процес навчання медіаграмотності навчання слід зробити більш інтерактивним. У цьому допоможе застосування на уроках робота-помічника з медіаграмотності, який зробить навчальний процес різноманітнішим. Він міг би взаємодіяти з учнями через графічний інтерфейс на екрані, а також синтезатор мовлення, і таким чином аналізувати текстові дані, давати учням пояснення щодо цього аналізу, і сприяти закріпленню учнями матеріалу, наприклад, пропонуючи їм вправи.

Інтелектуальна частина робота-асистента для виявлення пропаганди може бути розроблена завдяки сучасним технологіям машинного навчання. Зокрема, вирішення цього питання може бути запропоноване через такі інструменти, як великі мовні моделі. Завдяки своїй спеціалізації на роботі з текстами вони є конкурентним помічником у багатьох завданнях, які вимагають аналіз мови. Тому застосування великих мовних моделей для такого актуального завдання, як виявлення пропаганди, безсумнівно заслуговує на детальне дослідження.

У цій роботі описано концепцію інтелектуального робота-асистента з виявлення пропаганди на основі великих мовних моделей і розробку його інтелектуального компонента. Асистент повинен забезпечувати освітню функцію для потенційного застосування у навчанні медіаграмотності в інтеграції з роботом, і бути обладнаним зручним візуальним інтерфейсом для взаємодії з великою мовною моделлю, на якій він базується. Для розробки найефективнішого рішення потрібно дослідити декілька різних мовних моделей. Доцільно обрати найкращі з них за показниками на даний час, водночас які б відповідали наявним обчислювальним та фінансовим ресурсам.

Однією із потенційних складностей розробки є доволі містке завдання багатокласової класифікації технік пропаганди, комбінованої із ідентифікацією проміжків у тексті. Також буде досліджено вплив мови пропагандистських фрагментів (англійська, російська) на ефективність їх виявлення. Нарешті, потрібно розглянути показники доналаштованої моделі в порівнянні із базовою не дотренованою, щоб зробити висновок, наскільки ефективним є процес точного налаштування для даної задачі.

1 ЗАДАЧА РОЗРОБКИ ІНТЕЛЕКТУАЛЬНОГО РОБОТА-АСИСТЕНТА НА ОСНОВІ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ ДЛЯ ВИЯВЛЕННЯ ПРОПАГАНДИ

1.1 Постановка мети дослідження

Задача виявлення пропаганди є однією з ключових у сучасному інформаційному просторі, маючи широкий спектр застосувань. Пропагандистські матеріали можуть значно впливати на громадську думку, маніпулювати свідомістю людей та спотворювати реальну картину подій. Тому розробка інструментів для автоматичного виявлення пропагандистських технік є актуальною і важливою.

Найбільше хотілося б виділити освітню сферу, адже дуже важливо навчати підлітків та молодь, якої форми може набувати пропаганда і як легко її пропустити, читаючи текст, на реальних прикладах. Навчання молодого покоління медіаграмотності може забезпечити функціонування більш свідомого громадянського суспільства в Україні. Особливо важливим це є в контексті російсько-української війни і її інформаційного виміру. Тут і має місце застосування інтелектуального помічника, який проведе аналіз тексту на пропаганду, покаже місця, де саме вона виявлена, і проінформує про конкретні використані техніки пропаганди. Помічник може бути зінтегрований в освітнє середовище і бути прикладом інструменту для навчання.

Останні роки стали періодом стрімкого розвитку технологій на основі штучного інтелекту (ШІ), що значно розширило можливості обробки та аналізу текстової інформації. Впровадження трансформерних архітектур, таких як BERT, GPT та їхніх вдосконалених версій, відкрило нові горизонти у розумінні природної мови (NLP). Сучасні моделі можуть не лише класифікувати тексти чи визначати ключові елементи, а й генерувати складні, контекстуально релевантні відповіді, що наближає їх до рівня людського мислення. Таким чином, ці моделі активно впроваджуються у різноманітні сфери життя: від автоматизованого перекладу і персоналізованих рекомендацій до виявлення фейкових новин та боротьби з

дезінформацією. Також за рахунок їх наближеності до розуміння природної мови їх легко використовувати у формі розумних інтерактивних асистентів [2].

У контексті виявлення пропаганди технології штучного інтелекту дозволяють створювати більш точні та гнучкі рішення, які можуть адаптуватися до різних мов і культурних контекстів. Розробка інтелектуального робота-асистента на основі новітніх ШІ-моделей відкриває можливість не лише для автоматизованого аналізу текстів, а й для інтерактивної взаємодії з користувачами, що робить цей інструмент ефективним у освітньому та соціальному середовищах.

Попри існування рішень із виявлення пропаганди на основі нейронних мереж та моделей на основі трансформерів, не всі з них досліджують використання нових великих мовних моделей, і мало які враховують дані пропагандистських текстів на мовах інших, окрім англійської.

З урахуванням вищезазначеного, метою цієї роботи є підвищення ефективності виявлення та класифікації технік пропаганди у текстовому контенті, зокрема в освітніх цілях, за рахунок розробки програмного рішення з використанням технологій штучного інтелекту.

Для досягнення мети планується виконання таких кроків:

– проаналізувати наявні методи та реалізації виявлення пропаганди у текстах;

– зібрати вибірку даних з новинними текстами англійською та російською мовами, які містять та не містять пропагандистські техніки для навчання і тестування моделі;

– дослідити використання великих мовних моделей для задачі виявлення і класифікації пропагандистських технік і вплив точного налаштування на показники моделей. Визначити за метриками найефективнішу з них;

– розробити інтелектуального асистента з можливістю застосування у робототехніці для інтерактивного використання;

– оцінити ефективність рішення за обраними метриками.

Об’єктом дослідження є виявлення пропаганди у текстах англійською та російською мовами. Предметом дослідження є виявлення пропаганди у текстах за допомогою великих мовних моделей.

Аспектами наукової новизни у роботі виступають:

– порівняльний аналіз роботи точно налаштованих моделей GPT-4o mini та моделей сімейства Gemma /Gemma 2 з завданням виявлення пропаганди у текстах;

– використання цих моделей на завданні ідентифікації проміжків пропагандистського тексту;

– використання двомовного набору даних для використання LLM для виявлення пропаганди у текстах.

1.2. Опис задачі виявлення та класифікації пропаганди

Задача виявлення технік пропаганди, розглянута в роботі, складається із двох частин. По-перше, це ідентифікація пропагандистських технік, виявлених у тексті (мультикласова класифікація). По-друге, це виявлення конкретного фрагменту (інтервалу символів), де знайдено пропагандистську техніку, що можна віднести до задачі ідентифікації проміжків (Span Identification).

Задача багатокласової класифікації текстів має ряд можливих підходів до її розв’язання. Зокрема, це методи Bag-of-Words (представлення тексту через множину його слів, не враховуючи їх порядковість) і TF-IDF (Term Frequency-Inverse Document Frequency – метод, який використовує показники вагомості слів у тексті за їх частотою) в поєднанні з класифікаторами машинного навчання (наївний Байєс, метод опорних векторів, логістична регресія) [3]. Наївний Байєс, завдяки своїй ймовірнісній природі, може якісно працювати з малими текстами та забезпечує швидке навчання на обмежених обсягах даних. Метод опорних векторів (SVM) ефективно працює у випадках, коли потрібно розділити текстові дані з чіткими межами, а логістична регресія часто використовується як базовий підхід для бінарної та мультикласової класифікації.

У сучасних підходах до обробки природної мови застосовуються більш складні техніки, такі як вбудовування слів (word embeddings), які дозволяють перетворювати текст у векторне представлення, що зберігає семантичну інформацію. Такі методи, як Word2Vec, GloVe та FastText, забезпечують навчання словникових векторів на основі контексту слів у великих корпусах текстів.

Далі, згорткові нейронні мережі (CNN) можуть бути використані для вилучення локальних шаблонів у текстах, тоді як рекурентні нейронні мережі (RNN) і їх покращені версії, такі як LSTM (Long Short-Term Memory) та GRU (Gated Recurrent Unit), здатні ефективно захоплювати довгострокові залежності у послідовностях тексту. Трансформери, такі як BERT (Bidirectional Encoder Representations from Transformers), стали стандартом у багатьох NLP-задачах завдяки здатності моделювати контекст з обох боків кожного слова в тексті, що дозволяє отримувати більш точні результати [4].

Ідентифікація проміжків – це ще одне завдання зі сфери обробки природної мови, яке має на меті виявити конкретні фрагменти тексту, що містять певні характеристики чи патерни. Це завдання передбачає ідентифікацію та класифікацію суміжних проміжків токенів у тексті. Для його вирішення використовуються як класичні підходи, так і сучасні нейромережеві методи.

Ймовірнісні моделі, такі як умовні випадкові поля (CRF), часто застосовуються для задач послідовної класифікації. Рекурентні нейронні мережі (наприклад, LSTM) є ефективними для обробки послідовностей, оскільки вони зберігають інформацію про попередні слова у контексті. Сучасні трансформерні моделі, такі як BERT або RoBERTa, забезпечують високу точність за рахунок глибокого контекстуального розуміння тексту [5]. Вони можуть бути доналаштовані на конкретних наборах даних для вирішення задач ідентифікації проміжків.

Для вирішення наявної комбінованої задачі мультикласової класифікації та ідентифікації проміжків у роботі буде використано великі мовні моделі. Велика мовна модель – це модель для обробки природної мови, натренована на великому текстовому корпусі через техніки керованого та напівкерованого навчання. В

основі великих мовних моделей лежать штучні нейронні мережі з декодерною трансформерною архітектурою [6]. Великі мовні моделі є багатофункціональним інструментом для різних видів роботи із текстами, а також пропонують можливість точного налаштування їх під конкретне застосування.

1.3. Огляд наявних реалізацій

Використання моделей-трансформерів для виявлення пропаганди зустрічається у наукових працях, зокрема, [7]. Автори ставлять під сумнів підходи до виявлення пропаганди, де тексти маркуються лише як пропагандистські чи не пропагандистські, і прагнуть підійти до проблеми браку пояснюваності результату. Вони ставлять задачі мультикласової класифікації та ідентифікації проміжків для знаходження пропаганди у тексті та пропонують рішення за допомогою моделей на основі трансформера BERT (Bidirectional Encoder Representations from Transformers) із додаванням класифікаторного шару для кожного рівня гранулярності (обробка тексту, абзацу, речення тощо). Також автори створили корпус новинних статей, анотованих вручну на рівні фрагментів з вісімнадцятьма пропагандистськими технологіями.

Застосування BERT і нейронних мереж для задачі класифікації є доволі популярним підходом і зустрічається і у кількох інших працях. Так, автори [8] досліджують нейронні архітектури CNN, LSTM-CRF і BERT для вирішення завдань виявлення пропаганди на рівні речень і фрагментів і виділяють лінгвістичні, макетні та тематичні ознаки у текстах, на основі чого презентують багатогранулярну та багатозадачну нейронні архітектури для спільної задачі виявлення пропаганди. Крім того, у праці застосовані ансамблеві схеми, такі як мажоритарне голосування, релакс-голосування тощо, для підвищення продуктивності системи.

Автори [9] досліджують задачу мультикласової класифікації технік пропаганди через навчання великих мовних моделей на базі промптів (підказок) і використання файн-тюнінгу (донавчання). У цій роботі представлена проста задача

класифікації із додаванням інструкції «chain of thought» (ланцюжок думок), де модель повинна видати аргументацію наданої класифікації. Моделями, використаними в роботі, є зокрема представники сімейства GPT. В даній роботі автори також використали набір даних, який містить статті новин, позначені 14 пропагандистськими методами.

Існують також реалізації, які досліджують виявлення пропаганди не у новинних текстах, а на постах у соціальних мережах, таких як Twitter (X). Наприклад, праця [10] представляє набір даних, що містить слабо анотовані твіти, які містять методи дрібнозернистої пропаганди, і нейронний підхід для виявлення та класифікації пропагандистських твітів за цими дрібнозернистими категоріями. Авторське рішення включає багатовекторне представлення вхідних даних твітів, щоб прискорити процес навчання, виокремити різні аспекти вхідного тексту, включаючи контекст, сутності, їхні зв'язки та зовнішні знання і змоделювати їхню взаємодію. Ця робота також використовує трансформерну архітектуру BERT із додатковими шарами.

Ще одним боком підходу до виявлення пропаганди є її аналіз у мультимодальному вимірі. Автори [11] досліджують неординарне питання виявлення пропагандистських технік у мемах, створивши новий корпус із техніками, які можуть з’являтися у тексті, зображенні чи обох. В роботі використовується BERT, ResNet152 та різні надбудови та комбінації цих моделей.

Вищеописані реалізації для ідентифікації пропаганди використовують набори даних англійською мовою. Найбільш схожі за предметною областю дослідження (ті, які досліджують новинні тексти) або використовують обмежений перелік архітектур (BERT та його комбінації), або ж фокусуються в більшій мірі на завданні класифікації. Також ще не досліджено виявлення пропагандистських технік за технологіями LLM у соцмережі Telegram, зокрема російськомовних каналах.

Отже, дослідження великих мовних моделей для завдання виявлення пропаганди, так само як використання двомовного набору даних, має потенціал і буде реалізовано в цій роботі.

1.4. Формулювання вимог до розробки ПЗ

Внаслідок аналізу наявних підходів та реалізацій поставленої задачі, можна сформувати такі функціональні вимоги до розробки інтелектуального робота-асистента для виявлення пропаганди:

– асистент повинен приймати текст (у форматах .txt, .docx або через веб-інтерфейс) і підтримувати введення тексту різними мовами (мінімум англійська, російська);

– асистент повинен приймати текст (у форматах .txt, .docx або через веб-інтерфейс) і підтримувати введення тексту різними мовами (мінімум англійська, російська);

– асистент повинен бути здатним аналізувати наданий текст на наявність пропагандистських технік (вирішувати задачу класифікації із ідентифікацією проміжків);

– асистент повинен виводити список знайдених технік із зазначенням типу техніки та текстових фрагментів, де вона застосована;

– асистент повинен відображати знайдені техніки на веб-інтерфейсі (з підсвічуванням відповідних частин тексту);

– асистент повинен надавати довідку про техніки пропаганди разом з визначеннями та прикладами для кращого розуміння користувачем;

– повинна бути забезпечена можливість інтегрування системи асистента в робота-помічника, зокрема через API;

– повинна бути підтримка механізму донавчання великої мовної моделі в основі асистента на нових даних.

Також потрібно поставити ряд нефункціональних вимог:

– продуктивність. Час аналізу міні-текстів (новинних постів) не повинен перевищувати 10 секунд;

– надійність. Система повинна мати високий коефіцієнт доступності;

– зручність (usability). Інтерфейс має бути інтуїтивно зрозумілим як для технічних, так і для нетехнічних користувачів. Доступ до асистента повинен бути можливим з різних пристроїв;

– сумісність. Асистент повинен інтегруватися з існуючими системами через RESTful API;

– модульність. Архітектура має підтримувати можливість додавання нових функцій без значних змін у основному коді.

Для виконання вищезазначених вимог було вирішено створити веб-застосунок, адже він є незалежним від платформ, відповідно може бути модифікованим під використання на інших пристроях. Також веб-застосунок надає графічний інтерфейс, який є необхідним для використання інтелектуального асистента у освітніх та дослідницьких цілях.

Таке рішення можна буде потенційно інтегрувати у робототехнічну систему для розробки повноцінного робота. В такому разі технічна інтеграція мала б передбачити такі аспекти:

– обробка вхідного тексту. В разі використання робота повинно бути можливо зреалізувати доступ до асистента через API або локально. Для цього можна використати веб-застосунок із доступом до API, або, в разі локальної інтеграції, якщо робот оснащений ПК, застосунок можна запускати локально;

– власне робототехнічна платформа. Для інтелектуального асистента можна використати систему з дисплеєм та мікрофоном;

– забезпечення візуалізації та інтерактивності. Забезпечення функції виведення результатів аналізу на екран та/або голосового озвучування результатів через синтезатор мови.

Дане дослідження фокусується на розробці рішення власне інтелектуального асистента з потенційною можливістю інтеграції з фізичним роботом та підтримкою необхідних функцій, залишаючи робототехнічну частину для подальшої перспективи.

Повну діаграму варіантів використання (Use Case) інтелектуального асистента наведено у додатку Б. В той же час, для неї можна навести список акторів (ролей) та варіантів використання.

Актори:

– користувач (наприклад, учень). Основний користувач робота, який взаємодіє з ним для навчання медіаграмотності;

– система. Інтелектуальна система, яка включає асистента на основі великої мовної моделі для аналізу пропаганди, та засоби взаємодії з користувачем.

Випадки використання:

– аналіз тексту на пропаганду. Учень вводить або озвучує текст, який робот аналізує на наявність пропагандистських технік;

– виведення звіту про техніки. Система надає детальний звіт із переліком виявлених технік;

– візуалізація результатів. Система відображає знайдені техніки у тексті, підсвічуючи їх;

– надання навчальної інформації. Система асистента відображає довідку про техніки пропаганди та їх приклади.

Ці варіанти використання враховують освітній контекст застосування робота-асистента, роблячи систему більш інтерактивною.

У перспективі такий інтелектуальний асистент може включати більше функцій. Наприклад, у разі інтеграції з фізичним модулем, додання функції голосового вводу та голосових пояснень результатів від робота. Також для покращення освітньої функції можна додати інтерактивні вправи на знаходження пропаганди та дискусійні модулі в разі використання на шкільних уроках. Діаграму бізнес-процесів робота-асистента наведено у додатку В.

1.5 Висновки до розділу

В першому підрозділі було описано постановку задачі дослідження, разом з його об’єктом, предметом та новизною. Обґрунтовано актуальність проблеми

виявлення пропаганди і навчання медіаграмотності, і доцільність застосування великих мовних моделей для її вирішення. Було поставлено мету і визначено кроки її досягнення в роботі.

Далі було розглянуто загальні підходи з машинного навчання до вирішення задач класифікації текстів та ідентифікації проміжків: методи представлення тексту в комбінації з класифікаторами, токенізатори, вбудовування, види штучних нейронних мереж, які можуть працювати з текстами, а також окремо трансформерні архітектури.

У підпункті 1.3 наведено короткий огляд наявних досліджень на тематику виявлення пропаганди за допомогою технік машинного навчання. Деяку увагу приділено тим працям, які використовують моделі на основі трансформерної архітектури. Виділено праці, які досліджували пропаганду у текстах, а також в інших видах контенту, наприклад, мультимедійному. Зроблено висновок, що для цієї задачі є поле для подальших досліджень.

У підпункті 1.4 проводиться аналіз функціональних та нефункціональних вимог до інтелектуального робота-асистента з виявлення пропаганди, також враховано освітній контекст його застосування. Окрім того, було виділено варіанти використання системи, які увійдуть у діаграму прецедентів.

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ РОБОТА-АСИСТЕНТА ДЛЯ ВИЯВЛЕННЯ ПРОПАГАНДИ

2.1 Опис структури інтелектуального робота-асистента

Інтелектуальний робот-асистент для виявлення пропаганди на основі великих мовних моделей повинен складатися з таких логічних компонентів:

– контролер робота. Робототехнічний компонент, призначений для фізичної взаємодії з користувачем (наприклад, через голосовий зв’язок);

– інтерфейс взаємодії. Інструмент, через який користувач буде взаємодіяти з роботом, зокрема вносити тексти для аналізу. Також може передбачати інші функції, наприклад, відображення інформативного контенту. Його можна реалізувати у вигляді графічного інтерфейсу;

– механізм виведення. Інтелектуальна підсистема, що забезпечує аналіз текстів на наявність пропагандистських текстів. Робот взаємодіє з нею через REST API запити;

– велика мовна модель, на якій побудований механізм виведення;

– Text-to-Speech елемент. Забезпечує озвучення результатів аналізу та інший інтерактив з користувачем.

Інші елементи (наприклад, елемент Speech-to-Text, відповідальний за прийом голосового вводу від користувача) можуть додаватися в залежності від специфіки обраного робота та LLM. Узагальнену схему подано на рисунку 2.1, а повну версію наведено у додатку Г.

В наступних підрозділах буде розглянуто основні аспекти інформаційного забезпечення розроблення механізму виведення робота-асистента, великі мовні моделі різних сімейств, які ляжуть в його основу, а також специфіку механізму тонкого налаштування моделей, який буде застосовано.

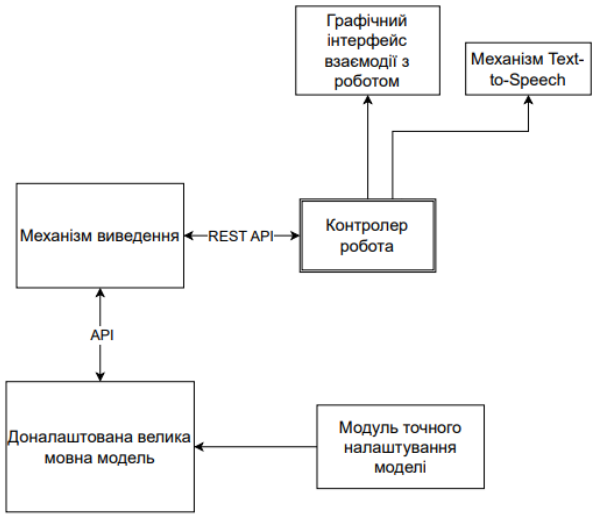


Рисунок 2.1 Загальна схема архітектури системи робота-асистента

2.2 Великі мовні моделі для виявлення пропаганди

В дослідженнях будуть використовуватися великі мовні моделі на основі трансформерної архітектури. Всередині трансформерної архітектури лежать такі компоненти [12]:

– вхідні вбудовування. Вхідний текст розбивається на одиниці (токени), зазвичай це слова, підслова або символи. Кожен токен перетворюється у векторне представлення у багатовимірному просторі, де слова з подібним значенням мають близькі вектори. Ці вектори навчаються або ініціалізуються за допомогою попередньо навчених методів, таких як Word2Vec, GloVe або спеціалізованих вбудувань для трансформерів (наприклад, BERT embeddings);

– позиційне кодування. Оскільки трансформери не мають вбудованої обробки послідовності, як рекурентні нейронні мережі, вони використовують позиційне кодування для врахування порядку слів. Ці кодування додаються до вхідних вбудувань і дозволяють моделі зберігати інформацію про позиції токенів у реченні. Вони можуть бути синусоїдальними або навчатися разом з іншими параметрами моделі;

– кодер. В серці трансформерної архітектури лежить багато кодувальних шарів, які аналізують вхідний текст і представляють його у вигляді прихованих змінних. Кожен шар кодера містить два основні компоненти. Механізм самоуваги дозволяє моделі визначати, на які інші слова слід звертати увагу при обробці кожного токена. Це допомагає виявити залежності між словами незалежно від їх відстані у тексті. Модель розраховує ваги уваги для кожного токена, що дозволяє оцінювати їхній взаємозв’язок у контексті. Нейронна мережа прямого поширення складається із повнозв’язаних шарів з нелінійними функціями активації, які приймають та обробляють кожен токен;

– декодерні шари. Забезпечують авторегресійну генерацію, при якій модель може генерувати послідовний вихідний результат з врахуванням попередніх токенів. Декодер також складається з декількох ідентичних шарів і відповідає за генерацію вихідного тексту. На відміну від кодера, він має додаткові механізми для обробки вже згенерованих токенів. Механізм маскованої уваги (Masked Self-Attention) забезпечує авторегресійну генерацію тексту, де кожен новий токен генерується з врахуванням попередніх токенів. Механізм уваги до виходу кодера (Encoder-Decoder Attention) дозволяє декодеру фокусуватися на релевантних частинах вхідного тексту, використовуючи приховані стани кодера;

– багатоголова увага (Multi-Head Attention). Механізм самоуваги виконується паралельно кількома головами уваги, що дозволяє моделі навчитися розпізнавати різні аспекти контексту одночасно. Кожна голова фокусується на різних частинах тексту чи на різних рівнях абстракції. Після обробки всі голови об'єднуються і проходять через лінійне перетворення;

– нормалізація шарів застосовується до кожного компонента архітектури для стабілізації навчання та покращення узагальнення моделі. Це допомагає уникнути проблеми вибухаючих або зникаючих градієнтів і сприяє швидшій конвергенції;

– вихідний шар. Останній шар моделі перетворює приховані представлення у вихідний текст або класифікаційні мітки. Для задач генерації тексту використовується softmax-шар, який визначає ймовірність наступного слова у

послідовності. У задачах класифікації застосовуються інші активаційні функції залежно від типу проблеми (наприклад, softmax для мультикласової класифікації або sigmoid для бінарної).

Описана структура LLM-моделі GPT (Generative Pre-trained Transformer, генеративний попередньо тренований трансформер) частково зображена на рисунку 2.2.

Рис. 2.2 Архітектура моделі GPT

Підсумовуючи, найбільш вагомим компонентом архітектури трансформера є механізм уваги, який дозволяє моделі встановлювати зв’язки між частинами даних, і таким чином імітувати розуміння природної мови [13].

2.2.1 Сімейство моделей GPT

Моделі GPT від OpenAI — це серія великих мовних моделей, побудованих на основі трансформерної архітектури, що призначені для генерації тексту, обробки природної мови та вирішення широкого спектра завдань NLP. Ці моделі продемонстрували значні досягнення у створенні контекстуально точного тексту,

відповіді на запитання, машинному перекладі, а також у різноманітних творчих завданнях.

Перша модель GPT була представлена у 2018р. компанією OpenAI, і відтоді цією ж компанією була випущена серія послідовно названих моделей, кожна наступна з яких перевершувала попередню завдяки більшій кількості параметрів, використаних у тренуванні:

– GPT (перше покоління). Започаткувало концепцію генеративних моделей на основі трансформерів;

– GPT-2. Відзначились значно більшим масштабом (1.5 мільярда параметрів) і здатністю генерувати текст високої якості, який важче відрізнити від людською. Модель стала популярною завдяки відкриттю її коду та можливості точного налаштування;

– GPT-3. Модель із 175 мільярдами параметрів стала проривом у генерації тексту, демонструючи глибоке розуміння контексту та багатозадачність. GPT-3 здатна виконувати різні завдання без додаткового навчання (zero-shot learning) або з мінімальним навчанням (few-shot learning);

– GPT-4. Найновіша версія відрізняється покращеною точністю, контекстною обізнаністю та здатністю працювати з більшими текстовими контекстами. GPT-4 демонструє високий рівень міркувань, розпізнавання складних шаблонів і адаптації до специфічних завдань.

Останніми розробками від OpenAI є GPT-3.5, GPT-4 та GPT-4o. Інформація про їх детальну архітектуру цих моделей недоступна широкому загалу, так само як інформація про текстові корпуси, на яких вони навчалися. Кількість тренувальних параметрів для GPT-3.5 складає 175 мільярдів, для GPT-4 оцінюється у 1.7 трильйона [14].

Великою мовною моделлю з сімейства GPT, яка використовується в роботі, є GPT-4o mini. Вона характеризується як найбільш економічно ефективна модель, будучи меншою та дешевшою за більшу GPT-4o, але розумнішою і такою ж швидкою, як попередня GPT-3.5. Модель має контекстне вікно на 128 тис. токенів, підтримує до 16 тис. вихідних токенів на запит і має знання до жовтня 2023 року.

Завдяки покращеному токенізатору, який використовується разом з GPT-4o, обробка неанглійського тексту цією моделлю є більш економічно ефективною, ніж з іншими моделями [15].

Фактором, який виступає на користь використання моделей сімейства GPT є їх доступність. Так, доступ до них можливий через API, і тому не потрібно витрачати власні обчислювальні ресурси на вміщення моделі.

2.2.2 Сімейство моделей Gemma

Одною із альтернатив сімейству GPT від OpenAI є великі мовні моделі Gemini та Gemma від Google Research.

Gemma - це LLM з відкритими для доступу вагами. Моделі Gemma створені на основі тих самих досліджень і технологій, що й більш потужні моделі Gemini. Архітектура цих моделей базується на вищеописаній архітектурі трансформерів з деякими доповненнями, і була вперше представлена у [16]. Наразі Google представили два покоління Gemma: Gemma та Gemma 2. Моделі першого покоління доступні в двох розмірах: модель з 7 мільярдами параметрів для розгортання і розробки на GPU і TPU, і модель з 2 мільярдами параметрів для CPU. Моделі другого покоління доступні в розмірах 2, 9 та 27 мільярдів параметрів.

На рисунку 2.3 зображено показники моделей Gemma на різних завданнях у порівнянні з конкуруючими LLM.

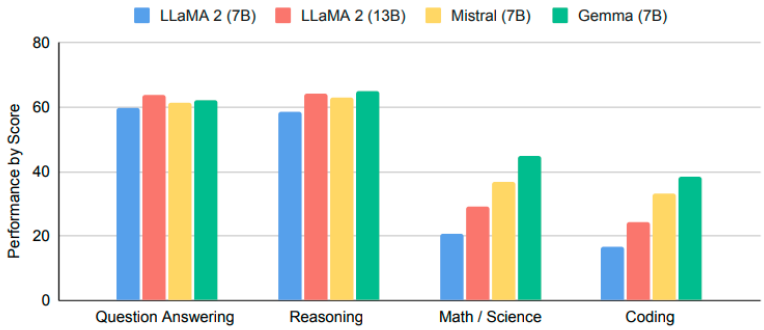


Рисунок 2.3 Порівняння Gemma з іншими LLM [17]

Ключовими доповненнями до раніше описаної архітектури GPT-моделей у випадку Gemma є:

– механізм багатозапитної уваги замість багатоголової у моделі 2B: за дослідженнями, цей механізм працює краще на малих масштабах;

– вбудовування RoPE (rotary positional embeddings, поворотні позиційні вбудовування) в кожному шарі, поділені між входом та виходом моделі, щоб зменшити її розмір;

– активації GeGLU. Стандартну нелінійність ReLU замінено на наближену версію функції активації GeGLU;

– RMSNorm. Вхід кожного підшару трансформатора, шару уваги та шару зворотного зв'язку нормалізується за допомогою RMSNorm для стабілізації навчання.

У другому поколінні моделей архітектура є подібною в більшості аспектів, відрізняються такі характеристики як чергування локальної уваги з ковзним вікном та глобальної уваги, пост-норма та пре-норма з RMSNorm, та увага через групові запити (Grouped Query Attention, GQA).

Ключові характеристики моделей Gemma та Gemma 2 наведено у таблиці 2.1.

Табл. 2.1. Характеристики моделей Gemma та Gemma 2

Параметри	Gemma 2B	Gemma 7B	Gemma 2 2B	Gemma 2 9B	Gemma 2 27B
Розмір вбудовування	2048	3072	2304	3584	4608
Шари	18	28	26	42	46
Вимір прихованих шарів прямого поширення	32768	49152	18432	28672	73728
Кількість голів	8	16	8	16	32

Кількість багатозапитних голів	1	16	4	8	16
Розмір голови	256	256	256	256	128
Розмір словника	256128	256128	256128	256128	256128
Дані тренування у токенах	3 трлн	6 трлн	2 трлн	8 трлн	13 трлн

Також до моделей Gemma було застосоване навчання на інструкціях – метод налаштування великих мовних моделей на маркованому наборі даних, що містить підказки (промпти) та відповідні результати. Це покращує продуктивність моделі не лише на конкретних завданнях, але й на виконанні інструкцій загалом, допомагаючи адаптувати попередньо навчені моделі для практичного використання [18].

З цього сімейства LLM в роботі буде використано найбільші моделі, з якими можна працювати з доступною потужністю – Gemma 7B (не дотренована на інструкціях) та Gemma 2 9B instruct (дотренована на інструкціях).

2.2.3 Донавчання LLM

Важливою характеристикою моделей на основі трансформерної архітектури є можливість їх адаптації до конкретного завдання, сфери чи стилю через донавчання (тонке настроювання, fine-tuning).

Донавчання або ж тонке настроювання – це підхід до передавального навчання, при якому параметри (ваги) попередньо натренованої нейронної мережі тренують на нових даних [19]. Моделі часто потребують точного налаштування для адаптації до нових словників, контекстів або завдань, тому що без цього передавальне навчання (використання моделі на специфічного формату контенті) дає гірші результати [20].

Моделі, попередньо навчені на великих текстових корпусах, тонко налаштовують, повторно використовуючи їхні параметри як відправну точку і додаючи специфічний для конкретного завдання шар, навчений з нуля. Альтернативним варіантом є точне настроювання всієї моделі, яке здатне дати кращі результати, але є більш затратним обчислювально.

У випадку з LLM-моделями, процес тонкого налаштування передбачає навчання наявних попередньо навчених LLM на специфічних для предметної області даних, тим самим покращуючи їхню здатність відповідати. Тонке налаштування має значення для адаптації великих мовних моделей до нових завдань або доменів без необхідності перенавчання з нуля, що призводить до підвищення економічної ефективності та зменшення обчислювальних витрат [21].

Тонке налаштування моделі є хорошим доповненням до навчання з кількох спроб. При навчанні з кількох спроб до інструкції для моделі додаються декілька прикладів, щоб уточнити, як саме виконувати завдання. А підхід донавчання дозволяє зекономити кількість токенів у інструкції, не включаючи приклади, і водночас отримати модель, яка вже володіє знаннями про специфічне завдання, поставлене їй [22]. Загальна схема тонкого настроювання зображена на рисунку 2.4, а в додатку Д розміщено структурну схему тонкого настроювання для моделей Gemma.

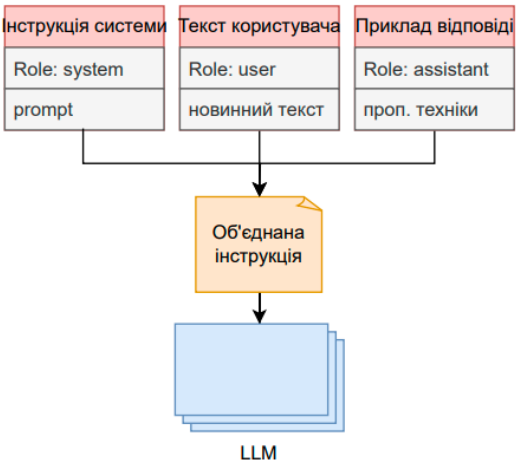


Рисунок 2.4 Загальна схема донавчання великої мовної моделі

Тому в роботі буде проведено експерименти з тонким настроюванням великих мовних моделей для оцінки, наскільки ефективним є цей підхід в контексті завдання виявлення пропаганди у текстах.

2.3 Дані для донавчання моделі на завдання виявлення пропаганди

2.3.1. Склад датасету

Важливим критерієм збору даних для вирішення задачі була наявність прикладів російською мовою. Оскільки система розробляється в контексті російсько-української війни та її інформаційного виміру, в даних повинні бути представлені приклади, які близькі до реалій пропаганди і які будуть ближчими для розуміння українському користувачу.

Вибірка даних для донавчання LLM-моделі, обрана для вирішення задачі, складається із двох частин. Перша, EMNLP (датасет названий через конференцію, на якій був представлений, Empirical Methods in Natural Language Processing), взята із праці [8] і містить 451 англомовну статтю, взяту із 48-ми новинних джерел. Кожному тексту відповідає список пропагандистських технік, виявлених у ньому, для кожної виявленої техніки присутній також інтервал в тексті, де її було виявлено. У цій частині вибірки присутні як пропагандистські, так і не пропагандистські статті. 82.5% статей, проте, взяті із джерел, які незалежні фактчекінгові організації визнали пропагандистськими, і ці статті мають тенденцію бути довшими.

Друга частина вибірки зібрана командою Mantis Analytics із російськомовних телеграм-каналів [23] і розмічена за таким самим принципом, як і EMNLP-датасет.

Загальний розмір вибірки – 1357 прикладів. Мінімальна довжина тексту – всього 10 символів, максимальна – 47706.

У первинному форматі датасет містив пари файлів з текстом та мітками до нього. Приклад фрагменту текстового файлу зображено на рисунку 2.5.

article699478811.txt (8.92 kB)

```
Iranian Aggression Intensifies

Last July, Major General Mohammad Bagheri, the Iranian Revolutionary Guards (IRGC) military commander and chief
The next day the Iranian regime warned of a "crushing" response if the United States were to designate the IRGC
President Trump has called the Iranian regime's bluff with his announcement last week that he would do just tha
Designating the IRGC as a terrorist organization and imposing new sanctions for its aggressive actions in the r
If Iran insists it can do what it wants militarily in terms of missile launches, support of terrorist groups su
The U.S. can impose sanctions against the Iranian regime's principal instrument for projecting aggressive, dest
The Iranian regime does not see it that way, however.
With the lifting of the nuclear-related sanctions making available billions of dollars to Iran's leaders to fur
Thus, it is threatening U.S. forces and bases in the region.
```

Рисунок 2.5 Фрагмент тексту з потенційною пропагандою

Кожен файл міток містив 4 колонки, розділені табуляцією: нумераційну з ідентифікатором файлу, колонку з назвою пропагандистської техніки та дві колонки з початковим та кінцевим символом фрази у тексті, де було виявлено цю техніку. Приклад такого файлу з мітками зображено на рисунку 2.6.

00028_02629	Repetition	41	45
00028_02629	Obfuscation,Intentional_Vagueness,Confusion	59	79
00028_02629	Repetition	334	339
00028_02629	Appeal_to_fear-prejudice	341	364

Рисунок 2.6 Приклад файлу з мітками

В подальших експериментах буде використано декілька варіацій датасету. Після первинної очистки даних, зокрема, знаходження дублікатів, видалення непотрібних символів тощо, цими варіаціями будуть такі:

– оригінальний датасет. Колонка Content та колонка manipulations. В останній записані пропагандистські техніки та номери початкового та кінцевого символів у тексті, де ця техніка виявлена;

– датасет для окремої задачі класифікації. Колонка Content та колонка manipulations. В останній записані лише пропагандистські техніки;

– другий варіант датасету для об’єднаної задачі класифікації та ідентифікації проміжків. Колонка Content та колонка manipulation_phrases. В останній містяться техніки та, замість символів, фрази з тексту, які їх містять. Ця варіація введена, щоб оцінити, наскільки показники ефективності моделі будуть відрізнятися від випадку з використанням першого оригінального датасету.

На рисунку 2.7 зображений фрагмент із першого варіанту датасету.

	Content	manipulations
641	Минпромторг не исключает временных логистическ...	
451	В Харькове на данный момент наблюдается тревож...	Appeal_to_Authority\t125\t151\n

Рисунок 2.7 Фрагмент оригінального датасету після препроцесингу

2.3.2 Техніки пропаганди у датасеті

В об’єднаній вибірці даних присутні 18 технік пропаганди. Далі подано їх назви та короткий опис згідно з творцями першої частини датасету [8].

– навантажена мова (Loaded Language). Використання слів/фраз із сильним емоційним забарвленням;

– обзивання або навішування ярликів (Name Calling, Labeling). Навішування ярликів на об'єкт пропагандистської кампанії;

– повторення (Repetition). Повторення одного й того ж повідомлення знову і знову;

– перебільшення або применшення (Exaggeration or Minimization). Або представлення чогось у надмірному вигляді: збільшення, покращення, погіршення, або створення враження, що щось є менш важливим або меншим, ніж воно є насправді;

– сумнів (Doubt). Ставиться під сумнів достовірність когось або чогось;

– апеляція до страху/упереджень (Appeal to Fear/Prejudice). Прагнення створити підтримку ідеї шляхом навіювання тривоги та/або паніки серед населення перед альтернативою;

– гасла (Slogans). Коротка та яскрава фраза, яка може містити ярлики та стереотипи;

– whataboutism. Прийом, який намагається дискредитувати позицію опонента, звинувачуючи його в лицемірстві без прямого спростування його аргументів;

– махання прапорами (Flag Waving). Гра на сильних національних почуттях для виправдання або просування дії чи ідеї;

– спотворення чиєїсь позиції (солом'яне опудало, Straw Man). Заміна тези опонента на подібну, яка потім спростовується замість початкової;

– причинне спрощення (Causal Oversimplification). Припущення єдиної причини або причини, коли насправді існує декілька причин проблеми;

– апеляція до влади (Appeal to Authority). Ствердження, що твердження є правдивим просто тому, що авторитетний орган або експерт з цього питання сказав, що воно є правдивим;

– кліше, що закінчують думку (Thought-terminating Cliche). Слова або фрази, які перешкоджають критичному мисленню та змістовному обговоренню певної теми;

– чорно-біла омана або диктатура (Black-and-white fallacy, dictatorship). Представлення двох альтернативних варіантів як єдино можливих, коли насправді існує більше можливостей;

– reductio ad hitlerum. Переконування аудиторії не схвалювати ідею, припускаючи, що ідея популярна серед груп, яких цільова аудиторія ненавидить;

– навмисна розпливчастість, плутанина (Obfuscation, Intentional vagueness, Confusion). Використання слів, які навмисно не є зрозумілими, щоб аудиторія могла мати власну інтерпретацію;

– надання несуттєвих даних (червона пляма, Red Herring): Представлення матеріалу, що не має відношення до обговорюваного питання, щоб відволікти увагу присутніх від суті справи;

– підтасовування фактів (Bandwagon). Спроба переконати цільову аудиторію приєднатися і діяти в тому ж напрямку, тому що «всі інші роблять те саме».

У об’єднаному датасеті присутній певний дисбаланс описаних технік. Найчастіше використовуваною технікою є навантажена мова, найрідше – підтасовування фактів. На рисунку 2.8 зображено кількість технік, які зустрічаються у тренувальній вибірці.

	manipulations	count
0	Loaded_Language	2621
1	Name_Calling,Labeling	1260
2	Exaggeration,Minimisation	927
3	Repetition	614
4	Doubt	536
5	Appeal_to_fear-prejudice	470
6	Appeal_to_Authority	348
7	Flag-Waving	322
8	Causal_Oversimplification	320
9	Slogans	223
10	Black-and-White_Fallacy	158
11	Thought-terminating_Cliches	110
12	Reductio_ad_hitlerum	90
13	Whataboutism	64
14	Red_Herring	55
15	Obfuscation,Intentional_Vagueness,Confusion	35
16	Straw_Men	22
17	Bandwagon	21

Рисунок 2.8 Підрахунок кількості технік у тренувальній вибірці

2.4 Метрики для оцінки моделі

Для оцінки моделей буде використано два підходи: окреме оцінювання класифікації технік пропаганди та окреме оцінювання ідентифікації проміжків із пропагандистськими техніками.

Під час фази тонкого настроювання (тренування та валідації) буде отримано метрики точності (accuracy) та втрат. Пізніше, під час фази виводу, буде пораховано метрики точності (precision), повноти (recall) та міри F1.

Точність (Precision) показує, яку частку об'єктів, передбачених як позитивні, модель визначила правильно:

$$Precision = \frac{TP}{TP + FP}, \tag{2.1}$$

де TP (True Positives) – кількість об'єктів, правильно класифікованих як позитивні; FP (False Positives) – кількість об'єктів, неправильно класифікованих як позитивні.

Повнота (Recall) показує, яку частку всіх реальних позитивів модель змогла правильно визначити.

$$Recall = \frac{TP}{TP + FN}, \tag{2.2}$$

де FN (False Negatives) – кількість об'єктів, які модель помилково класифікувала як негативні.

Міра F1 є середньозваженим значенням точності та повноти. Вона використовується, коли необхідний баланс між ними, особливо якщо дані є незбалансованими. Висока точність зазвичай знижує повноту і навпаки, тому F1 допомагає оцінити компроміс між ними.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}, \tag{2.3}$$

Для поставленого завдання будуть обчислені макросереднє та мікросереднє значення міри F1. Макросереднє обчислює метрику незалежно для кожного класу, а потім бере середнє значення (отже, ставиться до всіх класів однаково), тоді як мікросереднє об'єднує внески всіх класів для обчислення середньої метрики. Зокрема, у даному випадку буде присутня певна незбалансованість серед класів, тому більше значення буде мати мікросереднє значення.

Також буде обчислена оцінка F1 для кожного класу, що дасть уявлення про ефективність кожного конкретного методу пропаганди.

2.5 Висновки до розділу

В другому розділі детально описано компоненти інформаційного забезпечення розробки.

В підрозділі 2.1 зображено узагальнену схему системи робота-асистента з виявлення пропаганди на основі LLM. В наступному підрозділі описано архітектуру генеративного переднатренованого трансформера, на якій базуються усі великі мовні моделі, розглянуті в роботі, із фокусом як на теоретичних аспектах, так і на практичних. Розглянуто LLM, які будуть використані на практиці в роботі – це моделі сімейства GPT від OpenAI та Gemma/Gemma 2 від Google. Наведено інформацію про поняття точного налаштування, і яку користь воно приносить у вузькоспеціалізованих задачах.

Підрозділ 2.3 фокусується на розгляді двомовного об’єднаного набору даних, який використовується в роботі. Наведено його структуру, розмір, а також список технік пропаганди, присутніх у ньому. Підрозділ 2.4 описує метрики, які будуть використані для оцінки ефективності моделей.

3 РОЗРОБКА ІНТЕЛЕКТУАЛЬНОГО АСИСТЕНТА ДЛЯ ВИЯВЛЕННЯ ПРОПАГАНДИ

3.1. Вибір технологій та засобів

Для вирішення поставленої задачі потрібні засоби, які забезпечують зручну роботу з даними та великими мовними моделями, зокрема, обробку великих масивів текстів, форматування даних, взаємодію із моделями сімейства GPT через API, та з моделями Gemma безпосередньо через середовище. В той же час потрібен інструмент для створення мінімального користувацького інтерфейсу кінцевого продукту.

Основним інструментом розробки виступала мова Python.

Середовищем розробки виступали Jupyter Notebook, Kaggle та Visual Studio Code. Перше було обрано та використано переважно в першій частині розробки, для роботи з вибіркою даних: дослідження, очистка, форматування тощо. Kaggle Notebooks - хмарне обчислювальне середовище, яке забезпечує відтворюваний і спільний аналіз та підходить для виконання завдань машинного навчання. Очевидною перевагою цього середовища для мого завдання є можливість використовувати потужності GPU та TPU для роботи з великими моделями. Оскільки моделі Gemma великі за розміром, самої потужності CPU буде недостатньо. Середовище Visual Studio Code використовувалось для запуску користувацького інтерфейсу, тестування і оцінювання системи.

Для роботи з даними було використано бібліотеку Pandas, яка пропонує засоби для зручної роботи з великими вибірками даних, отримання їх статистики, обробки та форматування непідходящих рядків. В цій роботі використовувалась під час першої фази – збору та обробки датасетів з пропагандистськими текстами.

Для доступу до та донавчання моделі від OpenAI було використано програмний інтерфейс (API) Openai, пропонований самою компанією для взаємодії з їх моделями. За допомогою клієнта openai було встановлено зв’язок із базовою

моделлю, запускались fine-tuning jobs і проходили запити до готової натренованої моделі.

Для роботи з моделями Gemma було використано бібліотеки Keras, Tensorflow, keras-nlp та JAX. Бібліотека keras забезпечує підтримку розподіленого навчання моделей Gemma, використовуючи мульти-бекенд реалізацію, яка включає TensorFlow. За допомогою jax бібліотеку keras було налаштовано для доступу до TPU. Пакет keras-nlp було використано для тонкого налаштування та фази виводу.

Для оцінювання результатів, обрахунку метрик було використано бібліотеку scikit-learn, яка пропонує методи обрахунку метрик accuracy, precision, recall, F-1 score.

Для розробки користувацького інтерфейсу була застосована бібліотека streamlit, спеціалізований інструмент для веб-застосунків на основі даних. Ця бібліотека в тому числі забезпечує вимоги щодо потенційної інтеграції з робототехнічною системою. Наприклад, робот може взаємодіяти зі Streamlit-застосунком через REST API, або ж, якщо робот має вбудований ПК, як, наприклад, Raspberry Pi, застосунок можна запускати локально.

3.2. Система проведення експериментів

Для проведення експериментів в ході розробки та тестування моделі можна виділити такі кроки до виконання:

− вибір промпту (інструкції) для LLM (зокрема, буде випробувано спільну інструкцію для завдання класифікації та ідентифікації проміжків, а також окремо для класифікації);

− перетворення даних вибірки у коректний формат (різний для моделей OpenAI та Gemma);

− тонке налаштування моделей для двох завдань: перше одночасно класифікація та ідентифікація проміжків, друге – лише класифікація;

– проведення фази виводу (inference) та отримання метрик, за якими можна порівняти якість дотренованих моделей;

– порівняння отриманих моделей, вибір одної для інтегрування у кінцевий продукт – робота-асистента.

3.2.1. Вибір промпту для базової моделі

Початкова інструкція для моделі повинна вичерпно розкривати завдання, яке ставиться перед нею. Доцільно включити у промпт опис технік пропаганди, стислі приклади для кожної і бажаний формат виведення результатів.

Для кращого розуміння ефективності моделей буде використано два промпти: один для об’єднаного завдання класифікації та ідентифікації проміжків, інший лише для класифікації технік пропаганди. В свою чергу, було введено дві варіації промпту для об’єднаного завдання, оскільки для цього завдання було використано дві варіації датасету, описані в розділі 2. Фрагмент промпту для першого завдання (ідентифікація фраз) наведено на рисунку 3.1.

```
prompt_phrased = """You are a Text Classifier indetifying 18 Propaganda Techniques 
Loaded_Language - Using words/phrases with strong emotional implications to influen
Name_Calling,Labeling - Labeling the object of the propaganda campaign as either th
Repetition -  Repeating the message over and over in the article so that the audien
Exaggeration,Minimisation - Either representing something in an excessive manner or 
Appeal_to_fear-prejudice - Building support for an idea by instilling anxiety and/o
Flag-Waving; Playing on strong national feeling (or with respect to a group, e.g., 
Causal_Oversimplification -  Assuming one cause when there are multiple causes behi
Appeal_to_Authority - Stating that a claim is true because a valid authority or exp
Slogans - A brief and striking phrase that contains labeling and stereotyping, e.g.
Thought-terminating_Cliches -  Words or phrases that discourage critical thought an
Whataboutism - Discredit an opponent's position by charging them with hypocrisy wit
Black-and-White_Fallacy -  Giving two alternative options as the only possibilities
Reductio_ad_hitlerum - Persuading an audience to disapprove an idea by suggesting t
Doubt - Questioning the credibility of someone or something, e.g. 'Is he ready to b
Red herring - Introducing irrelevant material to the issue being discussed, so that
Bandwagon - Attempting to persuade the target audience to take the course of action
Obfuscation,Intentional_Vagueness,Confusion  - Using deliberately unclear words, so
Straw man - When an opponent's proposition is substituted with a similar one which 

For the given text please state which of the 18 propaganda techniques are present a
Loaded_Language: Corresponding phrase from the text,
Thought-terminating_Cliches: Corresponding phrase from the text,
Repetition: Corresponding phrase from the text

Here is the text:
"""
```

Рисунок 3.1 Фрагмент інструкції для об’єднаного завдання

Однакові пропмти використовувались для всіх моделей у дослідженні.

3.2.2. Перетворення даних для точного налаштування

Для донавчання вибірку було розділено на 3 піднабори: тренувальний, валідаційний та тестувальний. Тренувальний та валідаційний буде використано безпосередньо під час виконання fine-tuning job через OpenAI API, а тестувальний – для наочної оцінки роботи моделі і для розрахунку ширших метрик поза точністю та втратами. А саме, на тестувальному сеті буде проведено фазу виведення.

Тренувальний набір даних містить 1168 екземплярів, валідаційний – 349, тестувальний – 183.

3.2.2.1. Перетворення даних для GPT-4o mini

Для запуску процесу тонкого налаштування через OpenAI API дані потрібно перевести у формат JSONL (JSON Lines). Файл jsonl повинен складатися зі списку повідомлень, кожне повідомлення містить роль відправника та вміст [24]. Дозволеними ролями є система, асистент та користувач. Повідомлення системи зазвичай містить попередню інструкцію для моделі, яка має застосовуватися до кожного повідомлення користувача (в даному випадку це вищеописаний промпт із завданням), повідомлення користувача – те, що подається на вхід (в даному випадку текст із потенційною наявністю пропаганди), повідомлення асистента – бажана отримана відповідь від моделі (наявні техніки пропаганди і їх фрагменти).

Фрагмент із тренувального файлу, який подається на вхід моделі, зображений на рисунку 3.2.

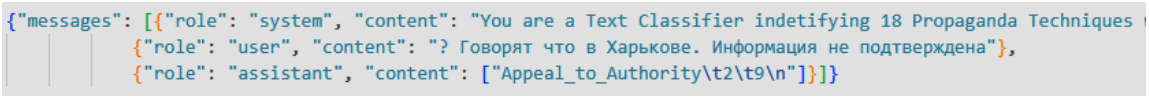


Рисунок 3.2 Приклад формату файлу з даними для донавчання

Після під’єднання до OpenAI API через індивідуальний токен і оголошення локального клієнта можна завантажувати тренувальні файли для препроцесингу. На рисунку 3.3 можна побачити програмне оголошення локального клієнта OpenAI , завантаження тренувального та валідаційного файлів для дотренування та фрагменти відповідей серверу, які містять дані про завантажені файли.

```
from openai import OpenAI
client = OpenAI()
```

```
client.files.create(
  file=open("dataset_train.jsonl", "rb"),
  purpose="fine-tune"
)
```

```
FileObject(id='file-8BhnIlJJqyJYg4hXLI9JcAr0', bytes=7447797,
status='processed', status_details=None)
```

```
client.files.create(
  file=open("dataset_val.jsonl", "rb"),
  purpose="fine-tune"
)
```

```
FileObject(id='file-XqU78p0Ob8VqE6qdWKkEdoGE', bytes=2167209,
atus='processed', status_details=None)
```

Рисунок 3.3 Завантаження тренувальних файлів

3.2.2.1. Перетворення даних для Gemma/Gemma 2

Для взаємодії із власними даними у середовищі Kaggle Notebook потрібно завантажити їх у власний Kaggle датасет, а потім імпортувати у конкретний ноутбук з кодом. На рисунку 3.4 зображено вигляд всіх імпортованих у ноутбук файлів у форматі csv.

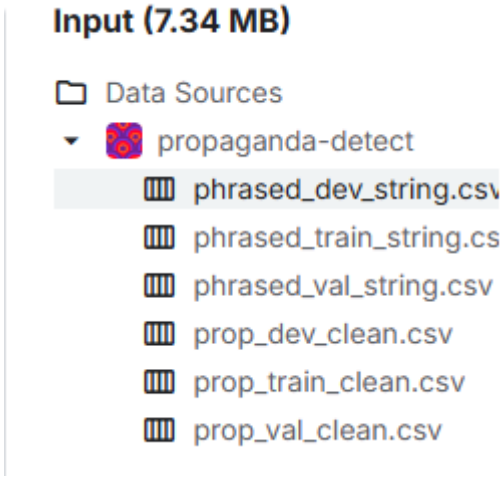


Рисунок 3.4 Вигляд завантажених у середовище файлів

Моделі Gemma та Gemma 2 є більш гнучкими в плані форматування даних для донавчання. Принциповим є лише однаковий формат тексту у тренувальних файлах та у тих, які використовуються під час фази виводу.

У даному випадку моделі буде подано масив даних, де кожен елемент вміщує інструкцію (промпт+новинний текст) та відповідь. На рисунку 3.5 зображено форматування датасету у вибраний формат.

```
df['instruction'] = prompt_phrased+' '+df['Content']
```

```
template = "Instruction:\n{instruction}\n\nResponse:\n{manipulation_phrases}"
data = [template.format(**row) for index, row in df.iterrows()]
```

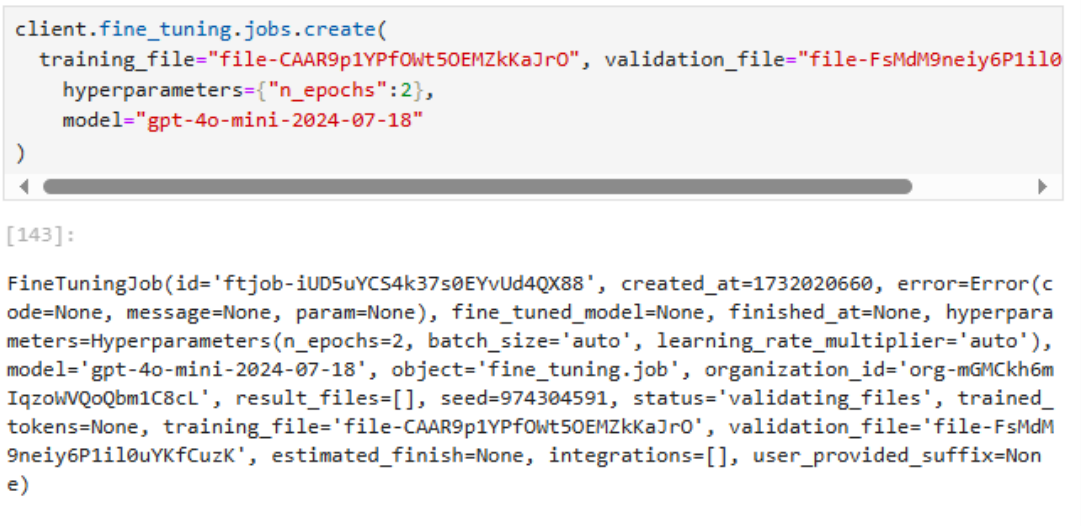
Рисунок 3.5 Форматування даних для моделі Gemma/Gemma 2

3.2.3. Донавчання моделі

3.2.3.1. Донавчання моделі GPT-4o mini

Процес донавчання запускається оголошенням задачі донавчання (fine-tuning job). Обов'язковими параметрами до запуску є ідентифікатор тренувального файлу, повернутий при його завантаженні на сервер, та ідентифікатор базової

мовної моделі. Є також ряд необов'язкових параметрів, серед яких ідентифікатор валідаційного файлу, кількість епох, розмір батчу, швидкість навчання тощо. Приклад оголошення задачі і відповідь сервера з її ідентифікатором зображена на рисунку 3.6.



```
client.fine_tuning.jobs.create(
  training_file="file-CAAR9p1YPfOWt5OEMZkKaJrO", validation_file="file-FsMdM9neiy6P1il0
    hyperparameters={"n_epochs":2},
    model="gpt-4o-mini-2024-07-18"
)
```

```
[143]:

FineTuningJob(id='ftjob-iUD5uYCS4k37s0EYvUd4QX88', created_at=1732020660, error=Error(c
ode=None, message=None, param=None), fine_tuned_model=None, finished_at=None, hyperpara
meters=Hyperparameters(n_epochs=2, batch_size='auto', learning_rate_multiplier='auto'),
model='gpt-4o-mini-2024-07-18', object='fine_tuning.job', organization_id='org-mGMCkh6m
IqzoWVQoQbm1C8cL', result_files=[], seed=974304591, status='validating_files', trained_
tokens=None, training_file='file-CAAR9p1YPfOWt5OEMZkKaJrO', validation_file='file-FsMdM
9neiy6P1il0uYKfCuzK', estimated_finish=None, integrations=[], user_provided_suffix=Non
e)
```

Рисунок 3.6 Оголошення процесу донавчання

Для даної задачі було порівняно декілька значень кількості епох, і як наслідок експериментів, вибрано параметр n_epochs = 2, оскільки подальше його збільшення ніяк не покращувало метрики тренування, а також щоб запобігти перенавчанню. Решту параметрів, якщо вони явно не вказані користувачем, внутрішній алгоритм OpenAI присвоює самотужки, беручи значення, найбільш підхожі до наданих даних.

Під час процесу точного налаштування можна також програматично отримати статус тренувальної задачі і список останніх подій для неї. Цю інформацію також можна отримати через інтерфейс користувача OpenAI API (рисунок 3.7).

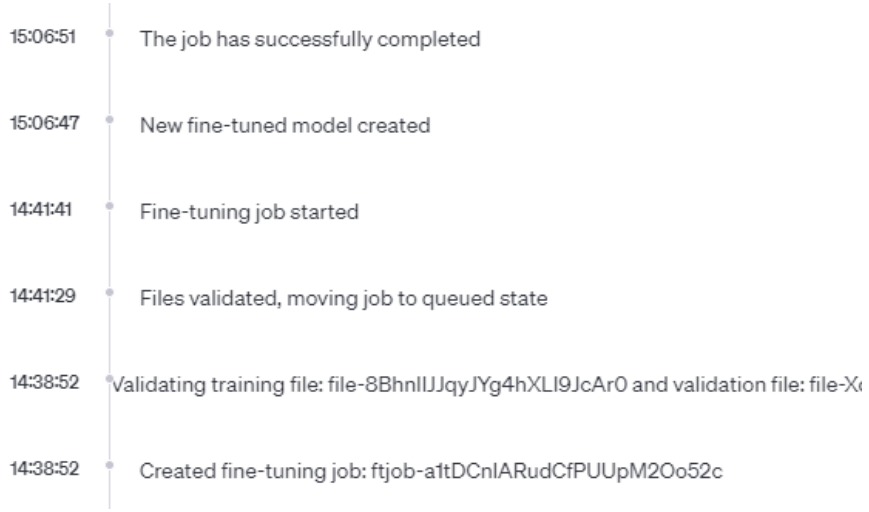


Рисунок 3.7 Події процесу донавчання в інтерфейсі OpenAI API

3.2.3.2. Донавчання моделей Gemma

Для проведення донавчання моделей Gemma буде недостатньо обчислювальних потужностей CPU (центральний процесор, Central Processing Unit) чи GPU (графічний процесор, Graphic Processing Unit), тому спершу потрібно змінити тип прискорювача на TPU (тензорний блок обробки, Tensor Processing Unit). TPU - це спеціалізовані процесори, розроблені Google для використання в машинному навчанні [25]. Вони призначені для швидкого виконання матричних обчислень, важливих для навчання та запуску нейронних мереж. Тож буде застосовано TPU для донавчання і виводу LLM. Kaggle Notebook надає квоту на безкоштовне використання TPU у розмірі 20-ти годин на тиждень, чого вистачає для тренування та тестування даної моделі.

Для застосування TPU потрібно обрати цей тип акселератора у середовищі Kaggle Notebook та розподілити пам’ять TPU на бекенді JAX. Програмне розподілення пам’яті зображено на рисунку 3.8.

```
os.environ["KERAS_BACKEND"] = "jax"
os.environ["XLA_PYTHON_CLIENT_MEM_FRACTION"] = "0.9"

jax.devices()
```

Рисунок 3.8. Програмне розподілення ресурсів TPU

В результаті отримуємо список із 8 віртуальних пристроїв TPU, які доступні (рисунок 3.9).

```
[TpuDevice(id=0, process_index=0, coords=(0,0,0), core_on_chip=0),
 TpuDevice(id=1, process_index=0, coords=(0,0,0), core_on_chip=1),
 TpuDevice(id=2, process_index=0, coords=(1,0,0), core_on_chip=0),
 TpuDevice(id=3, process_index=0, coords=(1,0,0), core_on_chip=1),
 TpuDevice(id=4, process_index=0, coords=(0,1,0), core_on_chip=0),
 TpuDevice(id=5, process_index=0, coords=(0,1,0), core_on_chip=1),
 TpuDevice(id=6, process_index=0, coords=(1,1,0), core_on_chip=0),
 TpuDevice(id=7, process_index=0, coords=(1,1,0), core_on_chip=1)]
```

Рисунок 3.9 Список доступних пристроїв TPU

Процес точного налаштування моделей Gemma відрізняється від процесу для моделей OpenAI, який можна проводити через API та обчислювальні ресурси компанії. У випадку з Gemma з моделлю доводиться працювати безпосередньо на наявній потужності, що не є простим завданням навіть із залученням TPU. Тому для полегшення процесу буде застосовано техніку LoRA.

LoRA (Low Rank Adaptation), або низькорангова адаптація, - техніка глибинного навчання на базі адаптерів, призначена для покращення точного налаштування великих мовних моделей, роблячи цей процес більш доступним. LoRA працює шляхом заморожування попередньо навчених ваг моделі та застосування матриць низького рангу, які, будучи доданими до базової моделі, створюють точно налаштовану модель [26]. З допомогою матриць низького рангу,

LoRA мінімізує необхідні обчислювальні ресурси, що робить можливим точне налаштування великих моделей на менш потужному обладнанні.

Крім того, для пришвидшення процесу буде запроваджено процес розподіленого тонкого настроювання. Зокрема, паралелізм моделі передбачає розподілення її ваг між кількома пристроями, горизонтальне масштабування і відповідно прискорення навчання.

Для реалізації процесу розподіленого навчання спочатку буде застосовано клас keras.distribution.DeviceMesh, який представляє кластер обчислювальних пристроїв, налаштованих для розподілених обчислень [27]. Зокрема, створимо DeviceMesh з формою (1, 8), щоб ваги моделі були розподілені між усіма 8 TPU (рисунок 3.10).

```
import keras
import keras_nlp

device_mesh = keras.distribution.DeviceMesh(
    (1, 8), ["batch", "model"], devices=keras.distribution.list_devices()
)
```

Рисунок 3.10 Створення device_mesh

Далі створимо макет layout_map, в якому буде вказано, як ваги і тензори повинні бути розбиті на шарди або репліковані за допомогою RegEx, і активуємо паралелізм моделі за допомогою device_mesh та layout_map. Фрагмент активації паралелізму зображено на рисунку 3.11.

```
model_parallel = keras.distribution.ModelParallel(
    device_mesh, layout_map, batch_dim_name="batch"
)

keras.distribution.set_distribution(model_parallel)
```

Рисунок 3.11 Активація паралелізму моделі

Після цього можна інтегрувати у середовище версію Gemma/Gemma 2 від Keras. Для цього потрібно попередньо додати цю модель в input секцію ноутбука, вибравши її із доступних моделей на Kaggle. Далі завантажуємо модель програмно (рисунок 3.12).

```
gemma_lm = keras_nlp.models.GemmaCausalLM.from_preset("gemma2_instruct_9b_en")
gemma_lm.summary()
```

Рисунок 3.12 Завантаження моделі Gemma 2 instruct 9b.

Можна отримати інформацію про завантажену модель, її параметри, за допомогою функції summary(). Фрагмент результату зображено на рисунку 3.13.

Layer (type)	Output Shape	Param #	Connected to
padding_mask (InputLayer)	(None, None)	0	-
token_ids (InputLayer)	(None, None)	0	-
gemma_backbone (GemmaBackbone)	(None, None, 3584)	9,241,705,984	padding_mask[0][0], token_ids[0][0]
token_embedding (ReversibleEmbedding)	(None, None, 256000)	917,504,000	gemma_backbone[0][0]

Total params: 9,241,705,984 (34.43 GB)

Trainable params: 9,241,705,984 (34.43 GB)

Non-trainable params: 0 (0.00 B)

Рисунок 3.13 Базова модель Gemma 2 instruct 9B

Також варто перевірити коректність шардингу моделі, щоб розподілене навчання відбулось успішно (рисунок 3.14). Зокрема, атрибут path в першій колонці показує шлях до шарду, shape в другій колонці показує розмір шарду.

```
<class 'keras_nlp.src.models.gemma.gemma_decoder_block.GemmaDecoderBlock'>
decoder_block_1/pre_attention_norm/scale                    (3584,)          PartitionSpec(None,)
decoder_block_1/pre_attention_norm/scale                    (3584,)          PartitionSpec(None,)
decoder_block_1/attention/query/kernel                      (16, 3584, 256)  PartitionSpec(None, 'm
odel', None)
decoder_block_1/attention/key/kernel                        (8, 3584, 256)   PartitionSpec(None, 'm
odel', None)
decoder_block_1/attention/value/kernel                      (8, 3584, 256)   PartitionSpec(None, 'm
odel', None)
decoder_block_1/attention/attention_output/kernel           (16, 256, 3584)  PartitionSpec(None, No
ne, 'model')
decoder_block_1/pre_ffw_norm/scale                          (3584,)          PartitionSpec(None,)
decoder_block_1/post_ffw_norm/scale                         (3584,)          PartitionSpec(None,)
decoder_block_1/ffw_gating/kernel                           (3584, 14336)    PartitionSpec('model',
None)
decoder_block_1/ffw_gating_2/kernel                         (3584, 14336)    PartitionSpec('model',
None)
decoder_block_1/ffw_linear/kernel                           (14336, 3584)    PartitionSpec(None, 'm
odel')
```

Рисунок 3.14 Шардинг моделі

Останнім кроком перед початком дотренування є компіляція моделі з додатковими параметрами: додаванням LoRA, вказуванням оптимізатора тощо. В ролі оптимізатора було використано варіацію Adam з кращою генералізацією та зменшеною тенденцією до перенавчання AdamW. Також можлива опція обрати тренувальні метрики для моделі. Фрагмент результату компіляції моделі з обраними параметрами у програмі зображений на рисунку 3.15.

```
Total params: 9,256,242,688 (34.48 GB)

Trainable params: 14,536,704 (55.45 MB)

Non-trainable params: 9,241,705,984 (34.43 GB)
```

Рисунок 3.15 Скомпільована модель

В порівнянні із базовою настройкою моделі, з додаванням конфігурації LoRA суттєво змінилась кількість тренованих параметрів: тепер вони займають

55.45 МБ пам’яті в порівнянні із загальною кількістю розміром 34.48 ГБ. Завдяки цьому є можливість переходити до наступного кроку, безпосереднього тонкого налаштування моделі.

Як базові параметри тренування було вибрано одну епоху та розмір пакету = 1. Також при тренуванні можна відстежувати прогрес у метриках (рисунок 3.16).

```
gemma_lm.fit(data, epochs=1, batch_size=1)
```

```
531/531 ━━━━━━━━━━━━━━━━━━━━ 432s 628ms/step - loss: 1.3302 - sparse_categorical_accuracy: 0.7152
```

Рисунок 3.16 Тренування моделі

3.2.4. Отримання початкових метрик та обчислення додаткових

За умовчуванням, OpenAI API пропонує автоматичний моніторинг тренувальних точності та втрат. Результати можна побачити через інтерфейс, або ж завантажити файлом формату csv.

На рис. 3.17 можна побачити графік втрат під час тренування.

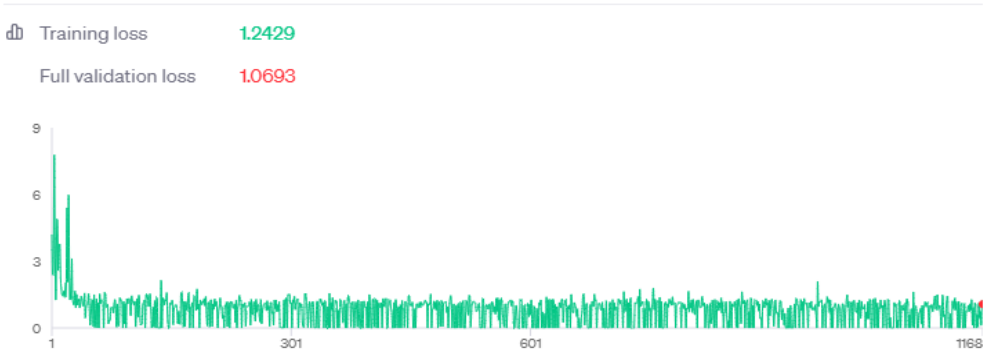


Рис. 3.17 Графік втрат

На рис. 3.18 зображено приклад числових метрик (тренувальних втрат та точності) із файлу csv.

```
step,train_loss,train_accuracy,valid_loss,valid_mean_token_accuracy
1,4.19525,0.53333,,
2,2.38483,0.66176,,
3,3.00622,0.62069,,
4,7.76879,0.66667,,
5,3.64901,0.53333,,
6,1.26207,0.71642,,
7,3.96846,0.5625,,
8,4.87241,0.5,,
9,2.58238,0.60714,,
10,3.619,0.41667,,
11,3.74186,0.6,,
```

Рис. 3.18 Обчислені метрики по донавчанню

Окремо будуть розраховуватися метрики під час фази виводу, описані у розділі 2.

Проведення фази виводу виконувалось окремо для всіх моделей, оскільки всі вони тренувалися у різних середовищах. Основною відмінністю у процесах було лише те, що при випробуванні моделей Gemma вихідні дані з результатами записувалися у папку хмарного середовища Kaggle Notebook, а при випробуванні GPT-4o mini – безпосередньо локально.

За обрахунок метрик у коді відповідає окремий файл calc_metrics.py. Оскільки використовувались три окремі підходи до оцінки перформансу моделі: окремо по задачі класифікації, окремо по задачі ідентифікації проміжків і по загальній задачі, то застосовувались окремі функції.

Під час іншого підходу оцінки моделі, тобто оцінки окремо задачі класифікації, буде використано готові бібліотеки метрик від scikit-learn. Для їх застосування потрібно перетворити список міток, повернутий моделлю, та список істинних міток у бінарний формат за допомогою класу MultiBinarizer пакету scikit-learn.

Приклад розрахунку метрик у коді зображений на рисунку 3.19.

```
# Macro-Averaged Precision, Recall, and F1-Score
macro_precision = precision_score(true_labels_binary, predicted_labels_binary, average='macro')
macro_recall = recall_score(true_labels_binary, predicted_labels_binary, average='macro')
macro_f1 = f1_score(true_labels_binary, predicted_labels_binary, average='macro')
file.write(f"Macro-Averaged Precision: {macro_precision}\n")
file.write(f"Macro-Averaged Recall: {macro_recall}\n")
file.write(f"Macro-Averaged F1-Score: {macro_f1}\n")
```

Рисунок 3.19 Обчислення додаткових метрик

Для задачі ідентифікації проміжків процес оцінки побудовано таким чином:

– Ітерування по файлах зі справжніми та передбаченими фразами.

– Для кожної передбаченої фрази пошук найкращого збігу з реальними фразами з точки зору схожості. Для порівняння використовується клас Difflib.SequenceMatcher – клас для порівняння послідовностей будь-якого типу, які піддаються хешуванню. Ідея полягає в тому, щоб знайти найдовшу суміжну підпослідовність, яка не містить «сміттєвих» елементів; ці «сміттєві» елементи - це ті, які нецікаві в певному сенсі, наприклад, порожні рядки або пробіли. Потім ця ж ідея рекурсивно застосовується до частин послідовностей ліворуч і праворуч від підпослідовності, що збігається [28].

– Якщо передбачена фраза збігається з фактичною фразою зі схожістю вище певного порогу, збільшуємо кількість істинних збігів на 1. Будь-яка невідповідність прогнозованої фрази є хибнопозитивним, а невідповідність фактичної фрази – хибнонегативним результатом.

– Підрахунок метрик точності, повноти та міри F1 з отриманими кількостями істинних, хибнопозитивних та хибнонегативних результатів.

Функцію, що відповідає за обрахунок подібності двох фраз, зображено на рисунку 3.20.

```
def calculate_similarity(phrase1, phrase2):
    """Compute similarity between two phrases using SequenceMatcher."""
    return SequenceMatcher(None, phrase1, phrase2).ratio()
```

Рисунок 3.20 Функція обчислення схожості

Для оцінювання, як добре модель справляється із двома завданнями одразу, було використано той самий підхід оцінки схожості, що й окремо для задачі ідентифікації. Таким чином, уже не лише кожна фраза, а кожна пара техніка-фраза оцінюється на схожість, і, в разі перевищення порогу, зараховується до істинних результатів.

Для вибору порогового значення схожості, за якого передбачений результат зараховується істинним, було проведено ручні експерименти з різними значеннями порогу і деякими текстовими даними, і суб'єктивно оцінено, наскільки такі фрагменти можна зараховувати до істинних. В результаті оптимальним значенням порогу було обрано similarity_treshold = 0.5. На рисунку 3.21 зображено фрази, які вважаються схожими з більшим чи рівним показником схожості, і тому потенційно можуть бути зараховані до істинних передбачень.

```
Predicted phrase:  и это лишь то, что попало в кадр  | Actual phrase:  И это лишь то, что попало в кадр!,
Predicted phrase:  представляем настоящую угрозу миру  | Actual phrase:  настоящую угрозу миру
Predicted phrase:  Будни киевского фольксштурма  | Actual phrase:  фольксштурма,
Predicted phrase:  ВОЕННОЕ ПРЕСТУПЛЕНИЕ ВСУ,  | Actual phrase:  ВОЕННОЕ ПРЕСТУПЛЕНИЕ ВСУ,
Predicted phrase:  ЗЕЛЕНСКОМУ ВЫСШАЯ МЕРА!!!,  | Actual phrase:  ЗЕЛЕНСКОМУ ВЫСШАЯ МЕРА!!!
Predicted phrase:  Самодержцу Всея Руси  | Actual phrase:  Ура Самодержцу Всея Руси,
```

Рисунок 3.21 Фрази з рівнем схожості більшим чи рівним 0.5

3.3. Інтелектуальний робот-асистент на основі LLM для виявлення пропаганди

Існує багато різного ступеня складності роботів, які можна застосовувати у даному випадку. Це може бути повнорозмірний Pepper Robot, який виступатиме помічником на класному занятті, невеликий Makeblock mBot з додатковим модулем дисплея, або різновиди конфігурацій роботів на основі RaspberryPi. Ці роботи зображені на рисунку 3.22.

Рисунок 3.22 Різновиди роботів Pepper Robot [29], Makeblock mBot [30], RaspberryPi [31]

Наприклад, Raspberry Pi є найбільш поширеним серед згаданих різновидів за рахунок своєї гнучкості – це одноплатний міні-комп’ютер, який відомий своєю гнучкістю та модулярністю, і широко застосовується в робототехніці, домашній автоматизації та автоматизації на підприємствах [31].

Наступною складовою системи робота-асистента для виявлення пропаганди є інтерфейс для взаємодії з користувачем – це точка контакту, через які людина може керувати роботом або отримувати від нього інформацію. Для даного випадку доцільним є використання графічного та голосового інтерфейсів. Графічний користувацький інтерфейс, в поєднанні із текстовим може включати в себе екран, сенсорні компоненти, поля для вводу, кнопки, а також, наприклад, компонент чату. А голосовий інтерфейс можна використати для озвучки певних вихідних повідомлень інтелектуального асистента (механізм Text-to-Speech).

Перетворення тексту в мовлення (TTS) - це здатність комп'ютера читати текст вголос. Механізм TTS перетворює написаний текст у фонематичне представлення, а потім перетворює фонематичне представлення у хвильові форми, які можна виводити у вигляді звуку [32]. На мові Python таку функціональність з озвученням різними мовами може надати бібліотека pyttsx3.

Для реалізації графічного інтерфейсу робота, залежно від його різновиду, можна використати певні бібліотеки мов програмування. Наприклад, Python

пропонує легку GUI-бібліотеку PyQt для створення кросплатформених графічних інтерфейсів, які можна використати у вбудованих системах, таких як RaspberryPi.

Наступним компонентом робота-асистента є інтелектуальна підсистема виведення на основі великої мовної моделі, яка буде забезпечувати обробку вхідних текстів і аналіз їх на пропаганду. У додатках И та К розміщено схеми інтелектуального асистента на основі моделей GPT-4o mini та Gemma. Буде розглянуто процес розробки такого інструменту для моделей від OpenAI.

Після донавчання модель стає доступною до використання у процесі виведення – стадія, коли модель готова до генерування бажаних відповідей. Для взаємодії з моделлю використовується Chat Completions API. Для створення відповіді існує метод create, в якому вказується ідентифікатор донавченої моделі, повідомлення, прийняте на вхід, та ряд необов'язкових параметрів, з якими можна експериментувати, наприклад, температура, ліміт вихідних токенів, кількість відповідей та ін. Приклад базового запиту на формування відповіді зображено на рисунку 3.23.

```
response = client.chat.completions.create(
        model=model_name,
        messages=[
            {"role": "system", "content": prompt},
            {"role": "user", "content": input_text}
          ]
    )
response_message = response.choices[0].message.content
return response_message
```

Рисунок 3.23 Функція для отримання відповідей від моделі

Початковий екран асистента виглядає таким чином (рисунок 3.24):

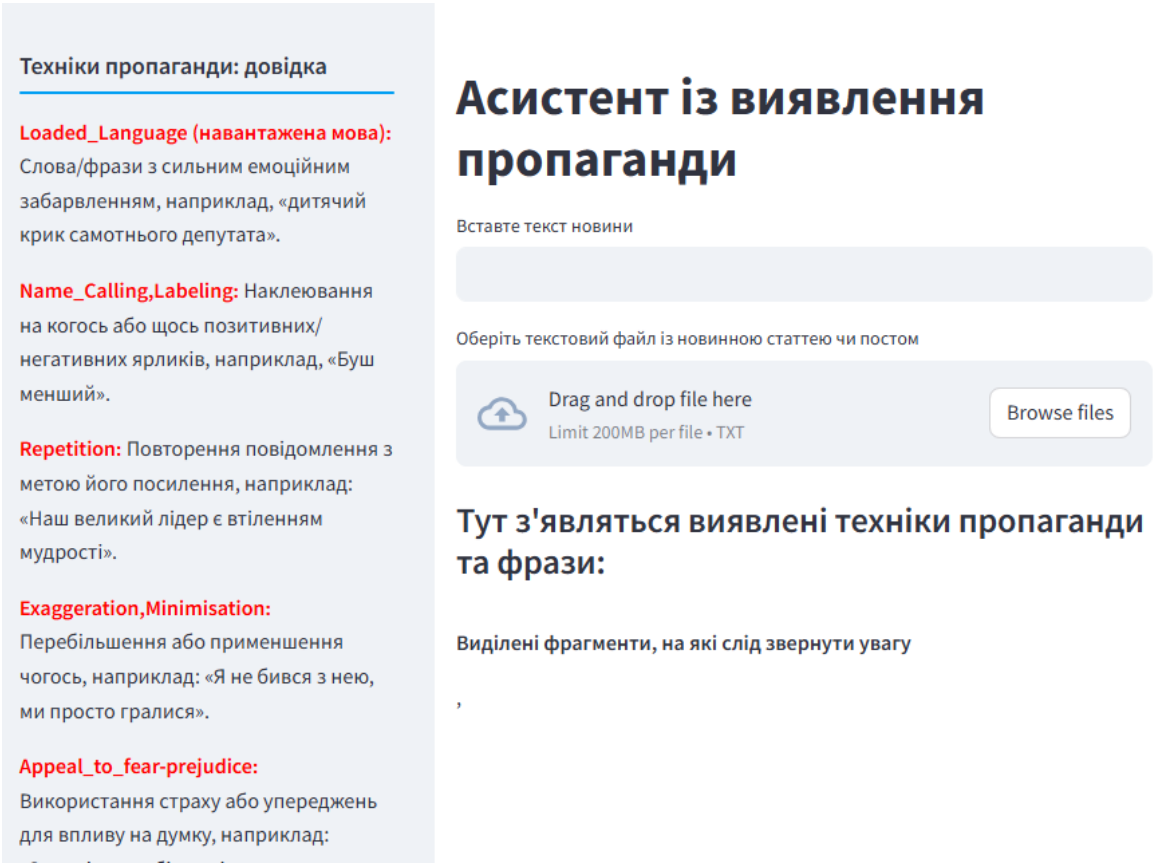


Рисунок 3.24 Початковий екран асистента

Присутня можливість введення тексту для аналізу, або ж завантаження файлу у форматі txt. Також наявна інформаційна довідка з переліком технік пропаганди, які зустрічаються у текстах, які транслюють пропаганду.

За допомогою бібліотеки streamlit дуже просто створювати візуальні блоки на веб-сторінці. На рисунку 3.25 зображено фрагмент створення початкової сторінки асистента.

```
st.set_page_config(page_title=' Detect Propaganda Easily!')
st.title('Асистент із виявлення пропаганди')
title = st.text_input("Вставте текст новини", "")

uploaded_file = st.file_uploader("Оберіть текстовий файл із новинною статтею чи постом",
                                 type=['txt'])
```

Рисунок 3.25 Створення початкової сторінки

Після завантаження текстового файлу, він обробляється донавченою моделлю. Тоді модель надає відповідь: спочатку список виявлених технік та фраз, після цього безпосередньо текст з підсвіченими фрагментами (рис. 3.26).

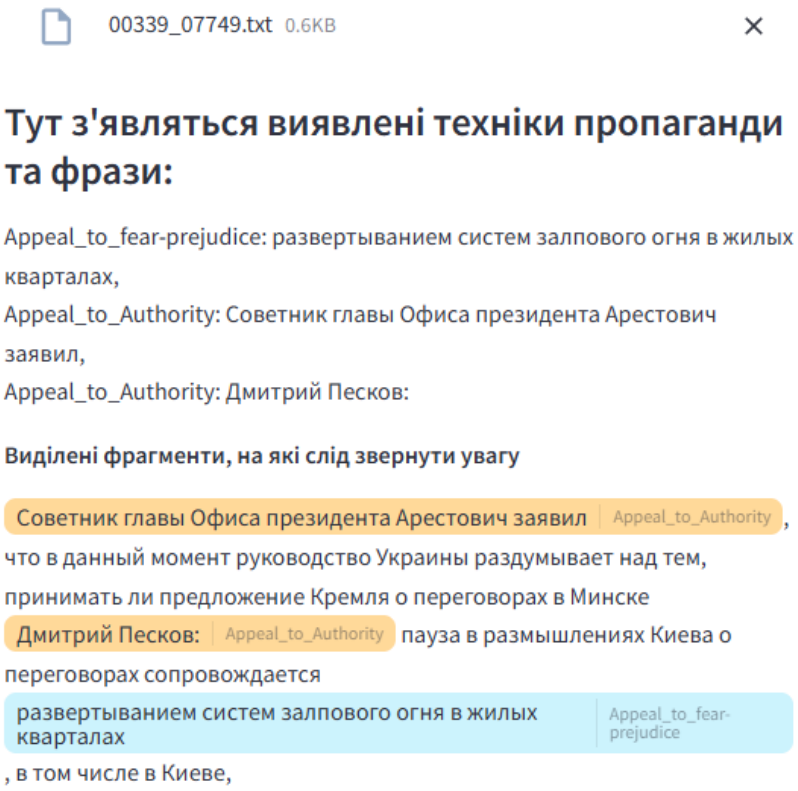


Рисунок 3.26 Відповідь моделі

Більше експериментів з вводом даних у модель буде представлено у розділі 4. Діаграму розгортання асистента із доналаштованою моделлю наведено у додатку Ж, а діаграму послідовностей у додатку Е.

3.4 Висновки до розділу

У третьому розділі було описано програмне забезпечення розробки інтелектуального робота-асистента з виявлення пропаганди. По-перше, було наведено перелік використаних в роботі технологій: середовища, мову

програмування, бібліотеки та пакети тощо, і обґрунтовано їх релевантність для даної задачі.

По-друге, було описано систему проведення експериментів, включно з формуванням промптів для моделей, попередньою обробкою навчальних даних (окремим для різних сімейств LLM), завантаженням даних для точного налаштування. Окремо було описано специфіки процесів точного налаштування. Так, наприклад, для моделей Gemma/Gemma 2 було наведено деталі щодо запровадження паралелізму і адаптера для швидшого і менш ресурсозатратного виконання, оскільки ця ресурсомістка модель повинна бути завантажена локально. Для моделей OpenAI було приділено увагу програмної взаємодії з ними через API. Окрім того, були описані детальніше метрики для експериментів з точки зору їх програмного обчислення.

По-третє, було приділено увагу безпосередньо вигляду інтелектуального робота-асистента, і того, яким чином він буде розгорнутий. Було наведено приклади взаємодії з ним.

4 РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТАЛЬНОГО ДОСЛІДЖЕННЯ ТА ПРАКТИЧНОГО ВИКОРИСТАННЯ

4.1. Постановка експериментального дослідження

Для дослідження та порівняльного аналізу великих мовних моделей у роботі буде проведено експерименти з різними моделями, доналаштованими на різних наборах даних, які були описані в розділі 2.

Для одночасного виконання завдань класифікації та ідентифікації проміжків було доналаштовано моделі GPT-4o mini та Gemma 2 9B, як дві найбільші з доступних за ресурсами. Для лише завдання класифікації було доналаштовано всі моделі. Наявні моделі та датасети, на яких вони були доналаштовані, наведені у таблиці 4.1.

Таблиця 4.1. Колонки: моделі, рядки: датасети, на яких вони були дотреновані

	GPT-4o mini, дотренована	Gemma 7B, дотренована	Gemma 2 9B, дотренована
Датасет лише з техніками (класифікація)	Так	Так	Так
Датасет з фразами (об’єднана задача)	Так		Так
Оригінальний датасет (об’єднана задача)	Так		

Загалом для кожного завдання метрики будуть рахуватися як окремо для задачі класифікації та задачі ідентифікації проміжків (для моделей, які були для цього донавчені), так і загально для об’єднаної задачі. Таким чином легше оцінити, з яким завданням LLM справляється краще та наскільки. Варто зазначити, що ці метрики будуть пораховані також для базової моделі GPT-4o mini, щоб виявити,

наскільки ефективним є застосування тонкого настроювання при даній задачі. У таблиці 4.2 розміщено розподіл моделей та метрики, які будуть для них пораховані.

Таблиця 4.2. Метрики задач, які будуть пораховані для LLM

	GPT-4o mini, дотренована		Gemma 7B, дотренована	Gemma 2 9B, дотренована	GPT-4o mini
	Температура=1	Температура=0.5			
Окремо метрики класифікації	Так	Так	Так	Так	Так
Окремо метрики ідентифікації проміжків	Так	Так		Так	Так

4.2 Результати експериментів

4.2.1 Результати експериментів для GPT-4o mini

Спочатку буде розглянуто метрики, обчислені при тренуванні моделі – це втрати та точність (accuracy).

Спочатку розглядається модель, дотренована на класифікаційній варіації датасету. На рисунках 4.1 та 4.2 зображено графіки втрат та точності при тренуванні даної моделі в залежності від кроків (метрики, надані автоматично сервісом OpenAI). Кроки – це певний еквівалент батчам, тобто точки, де була проведена перевірка. Для валідаційного процесу були надані метрики лише у певних контрольних точках, тому лінія на графіку зображена менш густою, ніж у тренувального процесу.

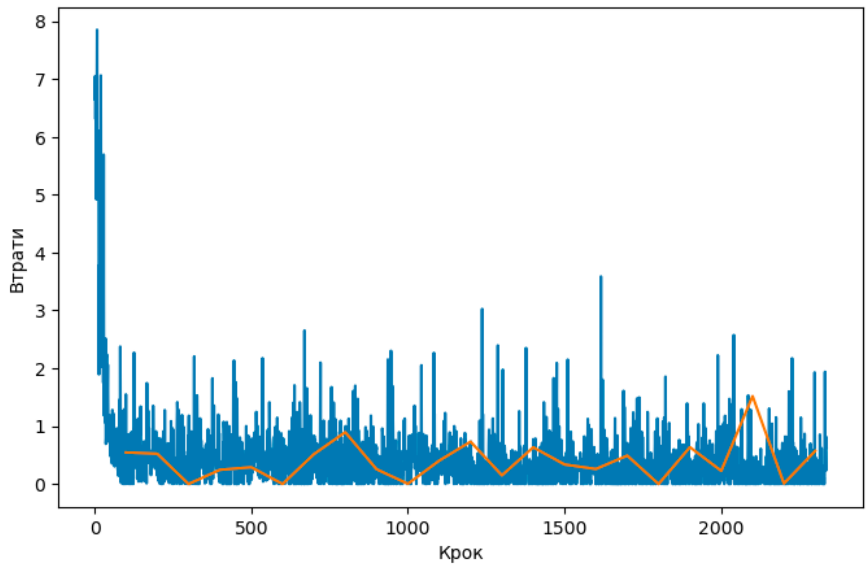


Рисунок 4.1 Графік втрат GPT-4o mini для класифікації.

Тут і далі густіші лінії – тренувальні дані, рідші – валідаційні дані.

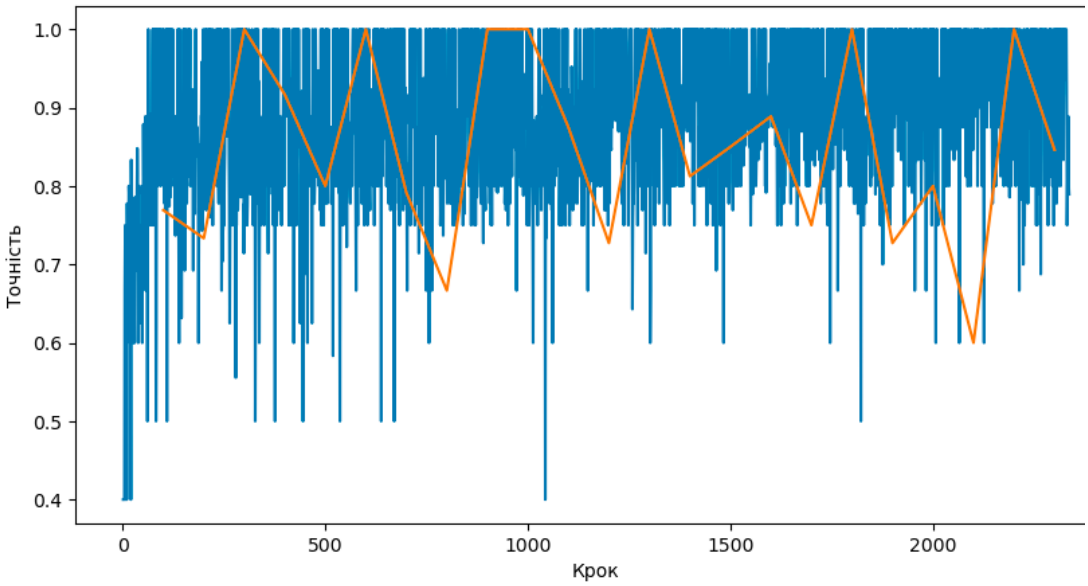


Рисунок 4.2 Графік точності GPT-4o mini для класифікації

По обох графіках видно, що після перших десятків кроків модель набрала тенденції до кращих метрик, однак не показала значного стрімкого росту чи падіння. Це, зокрема, доводить, що збільшення кількості епох недоцільне і не

приведе до значного покращення показників. На обох графіках валідаційні втрати та точність представлені без різких стрибків. Значення метрик втрат наприкінці тренування можна побачити через API OpenAI – зображено на рисунку 4.3.

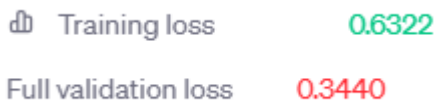


Рисунок 4.3 Втрати наприкінці тренування GPT-4o mini для класифікації

Наступна розглянута модель, яка була донавчена на датасеті з техніками та пропагандистськими фразами. Графіки втрат та точностей в залежності від кроків зображені на рисунках 4.4 та 4.5.

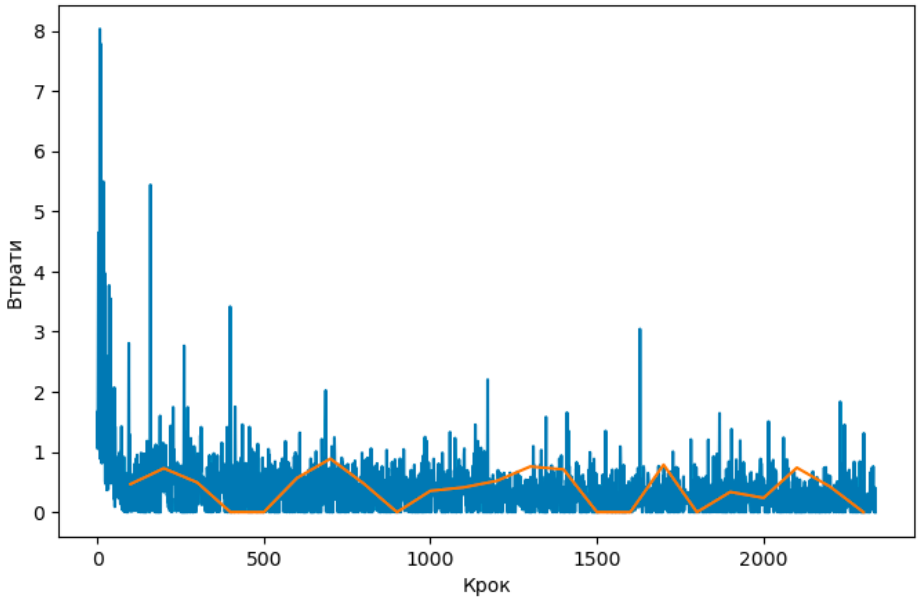


Рисунок 4.4 Графік втрат GPT-4o mini на датасеті з техніками та фразами

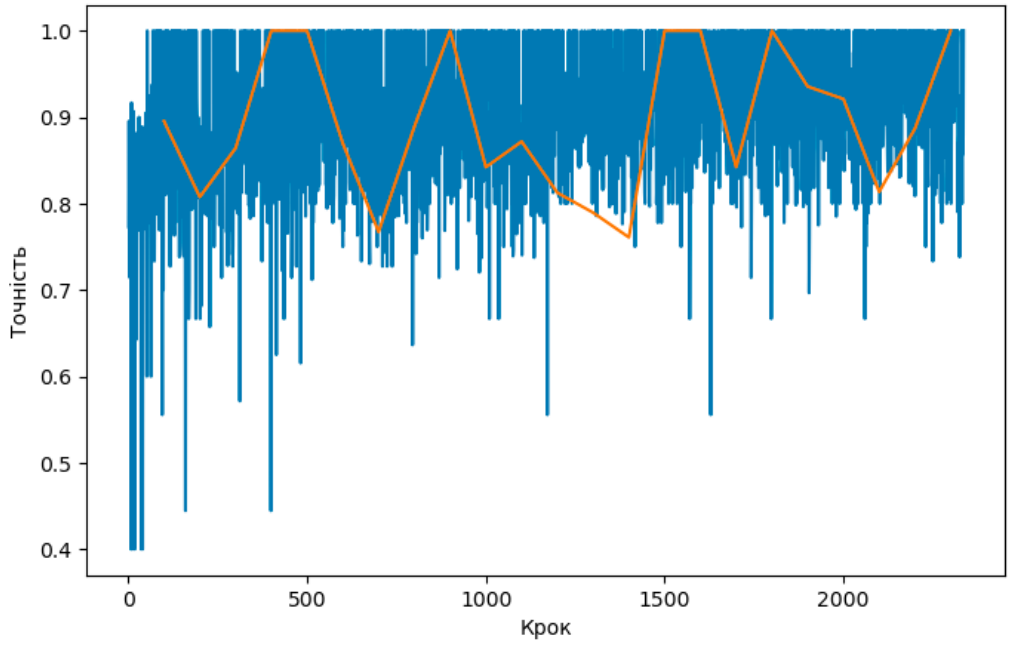


Рисунок 4.5 Графік точностей GPT-4o mini на датасеті з техніками та фразами

Тут тенденція схожа з попереднім випадком, хоча виглядає дещо оптимістичніше. Так, наприклад, валідаційна точність схильна до зростання у останніх контрольних точках, а графік валідаційних втрат виглядає пологішим.

Кінцеві втрати для варіації GPT-4o mini для ідентифікації технік та фраз зображено на рисунку 4.6.

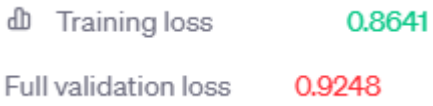


Рисунок 4.6 Втрати наприкінці тренування GPT-4o mini для ідентифікації технік та фраз

Останньою буде розглянуто LLM, доналаштовану на оригінальному датасеті із техніками та проміжками, зазначеними через номери символів. Для неї

наводиться лише графік втрат на рисунку 4.7 через несправність виведення метрик точності від OpenAI.

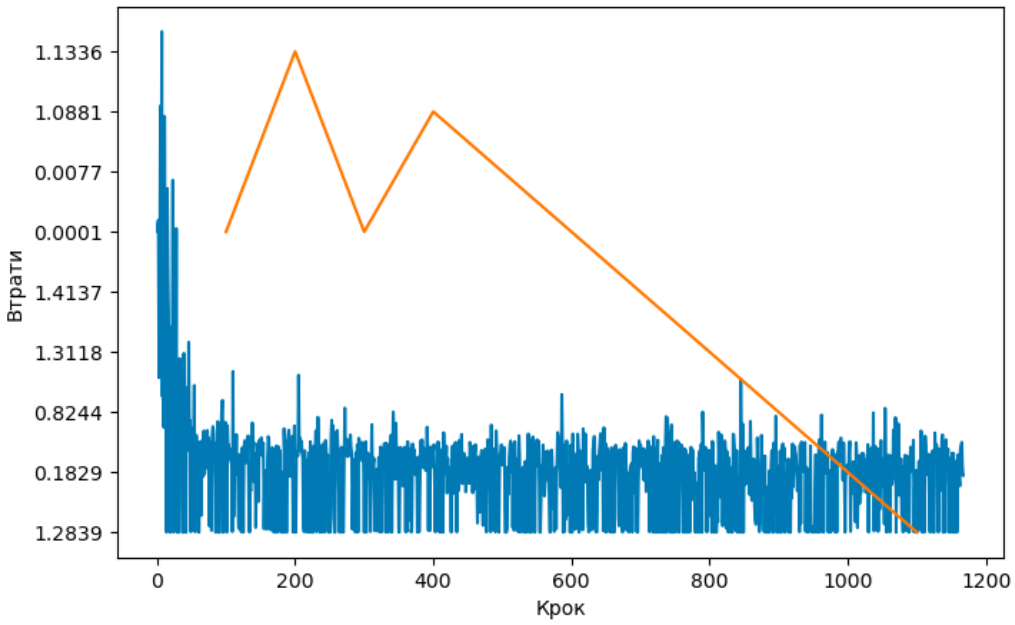


Рисунок 4.7 Графік втрат GPT-4o mini на датасеті з техніками та проміжками

Тут можна побачити цікаве стрімке лінійне падіння валідаційних втрат ближче до кінця тренування. Водночас тренувальні втрати не показують значної тенденції зменшення, залишаючись на стабільному рівні.

Загалом ці метрики, хоч і є частково інформативними, не показують наочної здатності LLM відповідати на потрібні запити. Тому було проведено експерименти з фазою виведення (inference), коли моделі надсилаються реальні запити. Під час цієї фази будуть використані інші метрики, раніше описані в розділі 2. В таблиці 4.3 наведені значення міри F1-мікроусередненої для варіацій, описаних у підрозділі.

Таблиця 4.3 Фінальні метрики F1 (мікроусереднене) для GPT-4o mini

	GPT-4o mini, класифікація + ідентифікація фраз		GPT-4o mini, класифікація + ідентифікація проміжків	GPT-4o mini, класифікація	GPT-4o mini
	Температур	Температур			

	a=1	a=0.5			
Окремо метрики класифікації	0.53	0.45	0.47	0.48	0.5
Окремо метрики ідентифікації проміжків	0.22	0.37	0.19	-	0.36
Метрики об’єднаного завдання	0.36	0.44	0.29	-	0.41

Видно, що найкращі результати для об’єднаного завдання і для завдання ідентифікації проміжків показує доналаштована модель GPT-4o mini з параметром temperature = 0.5. Для завдання класифікації найкращою є та сама модель, але з найвищою температурою.

Найгірші результати в середньому представила модель GPT-4o mini, класифікація + ідентифікація проміжків, яка була доналаштована на варіації датасету із початковими та кінцевими символами. Є припущення, чому так відбулося, і воно полягає в певній слабкості LLM в усіх завданнях, що стосуються обрахунку символів. Так, наприклад, у мережі нещодавно набув популярності так званий «полуничний тест» (strawberry test) [33], суть якого в тому, що при запитуванні у моделі, скільки літер «r» у слові «strawberry», модель відповідає неправильно, наприклад, кількістю 2. Це відбувається через те, що LLM не бачать текстові рядки у класичному вигляді, натомість бачачи лише результат після токенізації тексту. Через те, що видів токенізації може бути кілька (по словах, по підсловах, по символах), модель може не бачити слово через його прямий токен, а бачити, наприклад, два окремі токени і, відповідно, збиватися у рахунку символів. Основна мета LLM – розуміти і генерувати мову в контексті. Вони зосереджуються

на значенні слів та зв'язках між ними в більшій мірі, ніж на конкретних деталях, таких як підрахунок літер. Для операцій підрахунку символів вони не є спеціально оптимізовані.

Тому результат варіації моделі, натренованої на даних, які містять початкові та кінцеві символи, не є непередбачуваним. Модель показала кращі результати там, де замість нумерації були конкретні фрази з тексту.

Щодо різниці у параметрі температури, при практичному використанні моделі GPT-4o mini, класифікація + ідентифікація фраз помітні різні нахили у відповідях з температурою = 1 та 0.5. Перша дозволяє собі більш різноманітну класифікацію, часто з довшими фрагментами тексту. Це може бути корисно, якщо потрібні детальні інтерпретації тексту та ретельний його аналіз, розглянутий з різних сторін, і якщо важливо не пропустити щось. Однак часто такий аналіз не враховує контексту або підходить до визначення деяких категорій занадто інтенсивно, і в результаті виявляє фрази, які за своєю суттю не є пропагандистськими. На рисунку 4.8 зображено передбачення моделі з температурою 1 для одного з англомовних текстів. Видно, що модель ідентифікувала різноманітні класи, а деякі фрагменти є достатньо довгими. Водночас присутні кілька, принаймні на перший погляд, хибних тверджень, наприклад Loaded Language: grab.

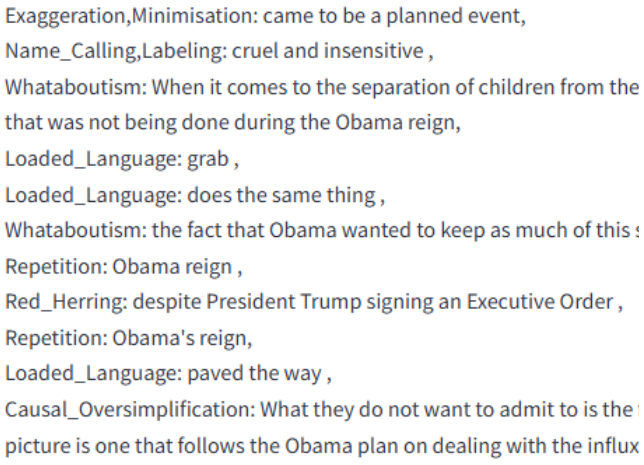


Рисунок 4.8 Передбачення LLM з температурою 1

В той же час, при температурі 0.5 поведінку моделі можна схарактеризувати як більш обережну у класифікації, але більш гранулярну при ідентифікації фраз. Тобто категорії стають менш різноманітними, з нахилами до тих, які зустрічалися частіше у тренувальних даних, а виявлені фрагменти пропаганди – коротшими. Ця тенденція підтверджується метриками повноти та точності (precision) для задачі класифікації, де точність вища за повноту (рисунок 4.9), а отже, модель схильна швидше пропускати істинні класи, ніж враховувати неіснуючі.

```
Micro-Averaged Precision: 0.6226415094339622
Micro-Averaged Recall: 0.3540772532188841
```

Рисунок 4.9 Метрики точності та повноти класифікації для моделі з температурою 0.5

Іншим спостереженням, поміченим при використанні температури 0.5, була дивна тенденція моделі до виведення кількох однакових рядків одночасно.

Для тої самої новинної статті, що й вище, модель з даним параметром вивела інший список, де збігаються лише кілька технік+фраз (рисунок 4.10).

```
Loaded_Language: literally over ,
Name_Calling,Labeling: the Obama reign,
Loaded_Language: the bottom line ,
Loaded_Language: to undermine progress ,
Loaded_Language: absurd,
Name_Calling,Labeling,
Name_Calling,Labeling: the Trump Administration ,
Name_Calling,Labeling: the Obama government ,
Name_Calling,Labeling: Leftists ,
Name_Calling,Labeling: the Department of Homeland Security ,
Loaded_Language: the Obama reign,
Name_Calling,Labeling: a carbon copy,
Name_Calling,Labeling: their idol, Obama ,
Name_Calling,Labeling: the Feds ,
Loaded_Language: nothing new ,
Loaded_Language: cruel and insensitive ,
Loaded_Language: a planned event,
Loaded_Language: quadruple ,
Loaded_Language: so much ,
Loaded_Language: speaks volumes ,
Loaded_Language: the Obama plan ,
Name_Calling,Labeling: the US government ,
Loaded_Language: no one can know ,
Loaded_Language: a carbon copy,
```

Рисунок 4.10 Передбачення LLM з температурою 0.5

Можна візуально порівняти ці передбачення з справжніми мітками у тестувальній частині датасету (рисунок 4.11).

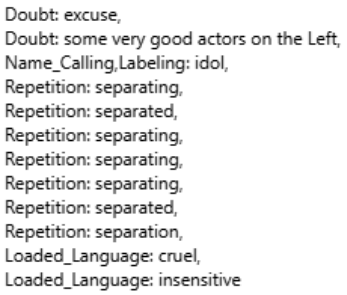


Рисунок 4.11 Справжні мітки

Видно, що обидві варіації моделі передбачили частково правильні пропагандистські фрагменти, але не всі, натомість визначивши кілька хибно позитивних.

Було також пораховано міри F1 для кожного класу окремо (рисунок 4.12).

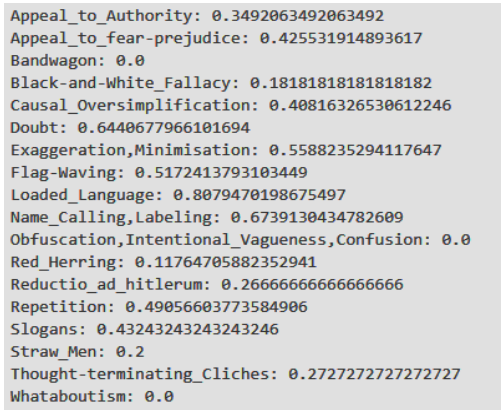


Рисунок 4.12 Метрики F1 для кожного класу окремо (класифікація, GPT-4o mini для класифікації та ідентифікації фраз)

Для двох класів цей показник становить 0, що означає, що модель не ідентифікувала ці техніки правильно. Однак, це не є чимось дивним, оскільки ці техніки є одними з найменш представлених у первинному наборі даних.

Також можна оцінити, наскільки краще або гірше справляється модель залежно від мови введеного тексту. Для цього буде взято доналаштовану GPT-4o mini для класифікації + ідентифікації фраз і оцінено метрики окремо для російської та англійської мови. Результати зображено у таблиці 4.4.

Таблиця 4.4 Метрики по мовах

Мова / F1 міра	Завдання класифікації	Завдання ідентифікації	Повне завдання
Англійська	0.48	0.31	0.38
Російська	0.42	0.19	0.19

Як видно, модель показує кращі результати для всіх завдань на англійських текстах. Це можна пояснити як і довшими текстами і більшою кількістю технік для них у англійській частині датасету, так і загалом кращої спеціалізації моделі GPT-4 o mini на англійській мові. Такі мови, як російська чи українська, більш токеномісткі (наприклад, одне слово російською частіше токенізується більш гранулярно, ніж одне слово англійською). Тому не доналаштовані моделі також часто показують гірші результати на іноземних мовах.

Можна помітити також цікаву тенденцію, що F1 міра для об’єднаного двомовного набору даних є вищою, ніж для кожного з них окремо. Для цього є кілька можливих причин.

Однією з потенційних причин є відмінності у розподілі класів. Нерівномірне представництво позитивних і негативних класів у кожній частині може бути наслідком зміщення балансу класів. В одній підмножині може бути непропорційно велика кількість складних прикладів, а в іншій може міститися менше прикладів класу меншини, що може знизити рівень повноти і зменшити точність. На противагу цьому, уніфікований набір даних зберігає початковий баланс класів і дозволяє моделі працювати більш послідовно і точно для всіх класів, покращуючи загальну оцінку F1.

Також у меншому наборі тестів варіативність і шум значно впливають на показники. В об'єднаному наборі даних ці ефекти можуть усереднитись, і забезпечити більш репрезентативну оцінку моделі. Так само уніфікований тестовий набір допомагає моделі краще узагальнювати і згладжує вплив складних випадків.

Моделі часто працюють краще, коли оцінюються на даних, які відображають різноманітність і складність всього проблемного простору, в даному випадку – новини з різних за своєю специфікою джерел, зокрема, вебсайти та Telegram-канали. Моделі могло бути складно адаптуватися до унікальних закономірностей або шуму в кожній частині, оскільки ці підмножини суттєво відрізняються за своїми характеристиками. Однак на уніфікованій множині модель піддається впливу повного спектру варіацій даних, що дозволяє їй більш ефективно узагальнювати і досягати вищих показників ефективності.

Загалом з цих результатів можна підсумувати, що інтелектуальний асистент на основі доналаштованої моделі дещо краще справляється із англійськими текстами для завдань ідентифікації проміжків, але для завдання класифікації результати приблизно рівні для обох мов.

4.2.2 Результати експериментів для Gemma/Gemma 2

Під час донавчання моделей Gemma було отримано попередні тренувальні метрики, за замовчуванням надані бібліотекою keras-nlp. Тут подано значення рідкої категоріальної точності, метрики, яка ділить кількість правильно класифікованих екземплярів на загальну кількість. На рисунку 4.13 зображено фінальні метрики для донавчання моделі Gemma 2 9B, тренованій на завданні класифікації + ідентифікації фраз.

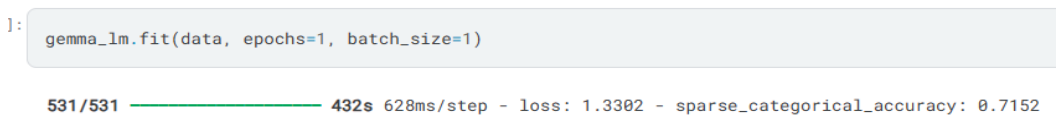


Рисунок 4.13 Метрики навчання моделі Gemma 2 9B

Видно велике значення втрат і порівняно високе значення категоріальної точності.

На рисунку 4.14 зображено фінальні метрики навчання для моделі Gemma 7B

.

```
gemma_lm.fit(data, epochs=1, batch_size=1)
```

```
1168/1168 ━━━━━━━━━━━━━━━━━━━━ 667s 522ms/step - loss: 0.7127 - sparse_categorical_accuracy: 0.8403
<keras.src.callbacks.history.History at 0x797de0111360>
```

Рисунок 4.14 Метрики навчання моделі Gemma 7B

Було зроблено висновок, що, попри достатні значення категоріальної точності, значення втрат не є низькими, і загалом, оцінювати роботу моделі буде ефективніше з використанням фази виведення.

Спочатку буде протестовано довільний екземпляр на моделі, зокрема спочатку варіації моделі Gemma 2 9B та Gemma 7B, доналаштовані на задачі лише класифікації. Зразок відповіді, який виявився у двох моделей однаковим, зображено на рисунку 4.15.

Після цього було проведено фазу виведення із випробуванням всіх тестувальних екземплярів на моделі. Цей процес зайняв доволі багато часу навіть попри використані техніки оптимізації ресурсів (4 години).

```
Here is the article/post:
?? Псаки - на вопрос об отправке войск на Украину: Задача президента США оце
нивать, что в интересах США, интересах национальной безопасности, интересах
американского народа и, безусловно, интересах мирового сообщества. Но вступа
ть в войну с Россией — не наш интерес. Поэтому, мы не отправим войска США на
Украину.


Response:
Appeal_to_Authority
```

Рисунок 4.15 – Відповідь доналаштованої моделі Gemma

Після проведення фази виведення було отримано метрики для моделей Gemma/Gemma 2, які були використані.

Таблиця 4.5. Метрики для моделей Gemma/Gemma 2

Метрики для моделей Gemma/Gemma 2	Gemma 2 9B, класифікація + ідентифікація фраз	Gemma 2 9B, класифікація	Gemma 7B, класифікація
Окремо метрики класифікації	0.30	0.45	0.41
Окремо метрики ідентифікації проміжків	0.24	-	-

На перший погляд значення метрик поступаються метрикам моделей GPT-4o mini, тому потрібно провести додаткове порівняння. Доцільно порівняти ще один з текстів з тестового набору (російською мовою) та відповідь моделі Gemma 2 9B (рисунок 4.16).

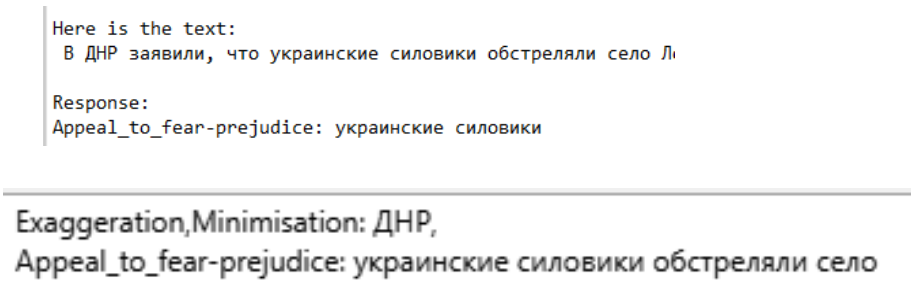


Рисунок 4.16 Згори – відповідь з тестувального набору, знизу – відповідь Gemma 2 9B

Також варто оцінити передбачення для текстів англійською мовою (рисунок 4.17). Ця модель також схильна до більш обережної класифікації, водночас трапляються повторення з різними техніками для одних і тих же фраз.

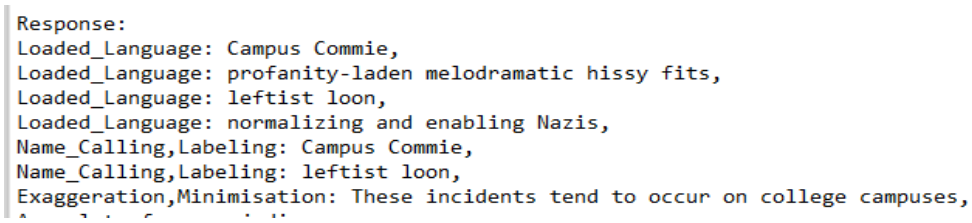


Рисунок 4.17 Передбачення Gemma 2 9B

Також потрібно оцінити на конкретному прикладі ефективність моделей 9B та 7B для окремого завдання класифікації. На рисунку 4.16 зображено істинні техніки одного з новинних екземплярів.

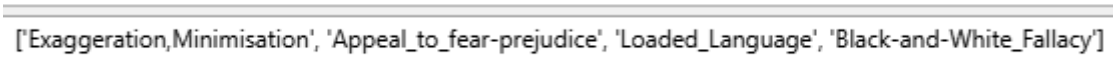


Рисунок 4.18 Приклад істинної класифікації

Доцільно порівняти відповіді моделей Gemma 7B та Gemma 2 9B відповідно (рисунок 4.19).

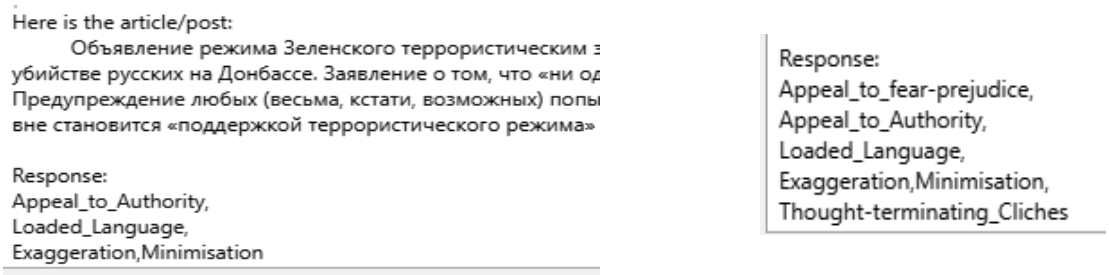


Рисунок 4.19 Відповіді Gemma 7B та Gemma 2 9B

Видно, що Gemma 2 9B дала більш класово різноманітну відповідь, одна жодна модель не захопила одну з істинних технік, Black and White Fallacy, яка, можливо, є дещо складнішою для аналізу.

По значеннях повноти та точності для класифікації з Gemma 2 9B очевидно, що модель схильна видавати більше хибно позитивних міток, ніж хибно негативних (рисунок 4.20).

```
Micro-Averaged Precision: 0.2521246458923513
Micro-Averaged Recall: 0.38197424892703863
```

Рисунок 4.20 Повнота та точність класифікації, Gemma 2 9B

Отримано також значення F1 окремо по класах (рисунок 4.21).

```
F1-Score per Class:
Appeal_to_Authority: 0.2988505747126437
Appeal_to_fear-prejudice: 0.30952380952380953
Bandwagon: 0.0
Black-and-White_Fallacy: 0.0
Causal_Oversimplification: 0.0625
Doubt: 0.13333333333333333
Exaggeration,Minimisation: 0.32432432432432434
Flag-Waving: 0.05555555555555555
Loaded_Language: 0.6188679245283019
Name_Calling,Labeling: 0.20588235294117646
Obfuscation,Intentional_Vagueness,Confusion: 0.0
Red_Herring: 0.0
Reductio_ad_hitlerum: 0.0
Repetition: 0.28169014084507044
Slogans: 0.11764705882352941
Straw_Men: 0.0
Thought-terminating_Cliches: 0.10362694300518134
Whataboutism: 0.0
```

Рисунок 4.21 Значення F1 по класах класифікації, Gemma 2 9B

За цими значеннями можна побачити більше нулів, ніж було в GPT-4o mini. Тобто, моделі Gemma таки більш схильні до певної «обережності» у класифікації, менше зважаючи на рідше представлені у тренувальному наборі класи.

В цілому можна підсумувати, що показники доналаштованих моделей Gemma/ Gemma 2 поступаються показникам конкурента GPT-4o mini у випадку мого завдання. Тому для застосування в інтелектуальному асистенті буде обрано варіацію моделі GPT-4o mini для завдання класифікації та ідентифікації фраз.

4.3 Порівняння результатів з попередніми дослідженнями

У багатьох із робіт, які досліджували тематику виявлення пропаганди, специфіка досліджень відрізняється від представленої у цій роботі. Наприклад, дослідження відбувались з іншими класами, з іншими видами текстів, або ж з іншими технологіями.

Одною з попередніх робіт є [8], де автори доналаштовують декілька LLM для датасету EMNLP, тобто англійської частини датасету з цієї роботи. Автори також виміряли метрики окремо для завдання ідентифікації проміжків і для спільного завдання, щоправда, без метрик класифікації. Показник F1 найефективнішої моделі з їх дослідження для об’єднаного завдання становить 22.58%. У даній роботі найкращий результат об’єднаного завдання становить 44%. Однак, варто зазначити різницю у підходах до обчислення метрик, оскільки фраза вважається правильно передбаченою, якщо вона відповідає певному порогу схожості. Тому, з певними налаштуваннями максимальної бажаної схожості, цей показник може зменшитись. Показник F1 для окремого завдання ідентифікації в авторів вищезазначеного дослідження становить 38% у найкращому випадку використання багатогранулярної моделі на основі BERT. У цій роботі найкращий показник для цього завдання становить 37% для доналаштованої на техніках і фразах GPT-4o mini з температурою 0.5. Це порівняно хороший результат.

У роботі [10] проводиться точне настроювання GPT-моделей для задачі класифікації. Можна порівняти їх результати з отриманими метриками для окремо класифікації. Показник F1 для натренованої GPT-4 у авторів становить 58%. У даному дослідженні найвище значення становить 53%. Це менший показник, але і сама трансформерна модель, використана у цій роботі, є меншою за розміром і більш економічно вигідною. Можливо, саме це стало одним з фактором більшої успішності моделі зі згаданого дослідження.

4.4 Результати практичного використання інтелектуального асистента з виявлення пропаганди

Для вимірювання ефективності та реальної корисності розробки потрібно також протестувати її на реальних прикладах.

Для такого тестування буде використано модифікацію моделі, яка показала найкращий рівень F1-міри для об’єднаного завдання класифікації та ідентифікації фраз, а саме, GPT-4o mini, класифікація + ідентифікація фраз.

Доцільно протестувати роботу цієї доналаштованої моделі спочатку на текстах з пропагандистських Telegram-каналів (тобто таких, які вважаються недостовірними в Україні). Оригінал такого тексту можна побачити на рисунку 4.22.

Рисунок 4.22 Приклад потенційного пропагандистського тексту

Цей текст заноситься у застосунок, можна у вигляді файлу або текстового фрагменту, та отримується відповідь моделі (рисунок 4.23).

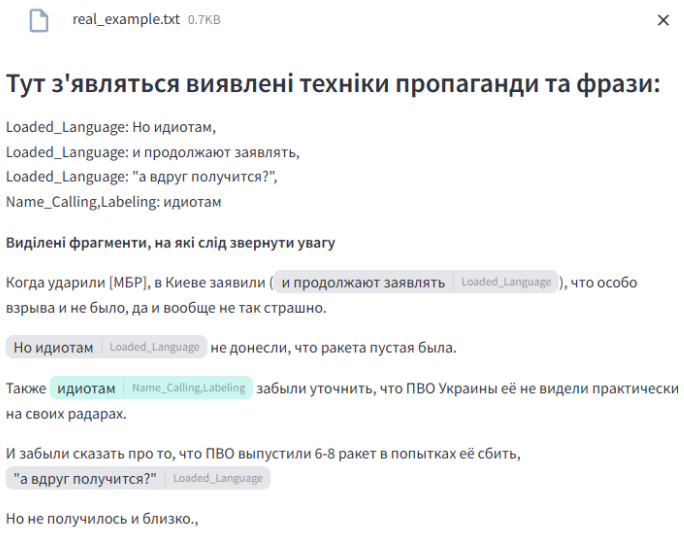


Рисунок 4.23 Відповідь моделі у застосунку

Видно, що LLM впоралась із завданням класифікації та ідентифікації, оскільки користувачеві пропонується звернути увагу на конкретні фрази, які справді видаються емоційно забарвленими.

Потрібно перевірити роботу моделі на довших текстах англійською мовою. Цього разу новинний текст було взято із одного з іноземних сайтів, визначеного як пропагандистський Центром стратегічних комунікацій та інформаційної безпеки [34]. Взяту новинну статтю зображено на рисунку 4.24.

Рисунок 4.24 Фрагмент англомовної потенційно пропагандистської статті

В цьому варіанті модель також справляється із завданням, відповідь показано на рисунку 4.25.

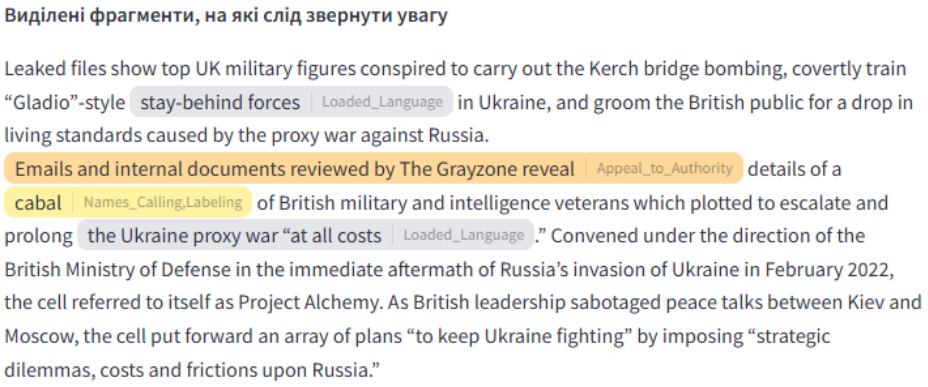


Рисунок 4.25 Виконання завдання на англомовному тексті

Важливо також, щоб модель не відзначала будь-який текст пропагандистським, навіть якщо він взятий з такого джерела. Перевірка зображена на рисунку 4.26.

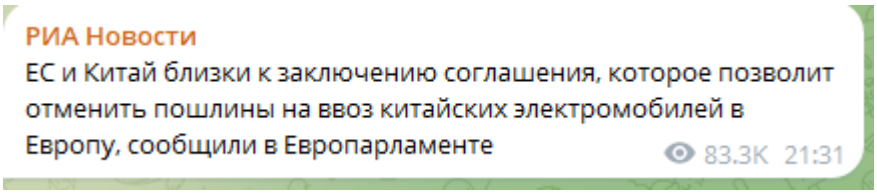


Рисунок 4.26 Фрагмент тексту з пропагандистського джерела

Відповідь моделі зображена на рисунку 4.27.

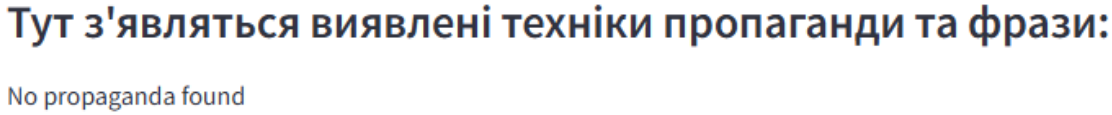


Рисунок 4.27 Відповідь моделі

Модель не показує пропагандистських фрагментів у цьому тексті, що є позитивним результатом, оскільки він таким явно і не є.

4.5 Висновки до розділу

В 4-му розділі було описано результати експериментальних досліджень задачі виявлення пропаганди у текстах за допомогою великих мовних моделей.

Було наведено список всіх варіацій LLM, які використовувались, з вказаними варіантами наборів даних, на яких ті доналаштовувались, і деякими іншими параметрами. Також було описано всі підходи до обрахування метрик на згаданих моделях.

Окремо для моделі GPT-4o mini та сімейства моделей Gemma/Gemma 2 було описано процес доналаштування, наведено тренувальні метрики та певні графіки втрат та точності. Також окремо для цих моделей було описано результати фази виведення разом з її метриками, і проаналізовано фактичну суть цих результатів. Було зроблено висновки по тренуванню та обрано варіацію моделі для інтегрування у інтелектуальний асистент.

Також було взято декілька прикладів реальних даних з пропагандистських джерел не з набору, і протестовано рішення на них.

У висновку можна стверджувати, що доналаштування моделі було виконано успішно, і модель може бути застосована у інтелектуальному асистенті для виявлення пропаганди в новинах.

5 РОЗРОБКА СТАРТАП-ПРОЄКТУ

5.1 Опис ідеї стартап-проєкту

Маніпуляції, фейкові новини та пропагандистські техніки активно використовуються для впливу на громадську думку, викривлення фактів та посилення соціальної напруги. Особливо це актуально в контексті війни, розпочатої Росією проти України, та решти глобальних криз, військових конфліктів та політичних кампаній. При цьому багато людей, особливо молоді, не володіють навичками критичного мислення для самостійного розпізнавання пропаганди. Розробка системи, яка автоматично виявляє пропагандистські техніки в текстах, є важливим кроком для підвищення рівня медіаграмотності молодого населення. Інтеграція асистента у навчальний процес дозволить учням і студентам краще розуміти, як працює пропаганда, та формувати навички аналізу текстів. Це особливо важливо для підготовки нового покоління свідомих громадян, здатних протистояти інформаційним загрозам. Розробка такого асистента також має ряд інших потенційних застосувань, представлених у таблиці 5.1.

Таблиця 5.1.

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Інтелектуальний помічник з виявлення пропаганди у новинних текстах	Освітній: навчання медіаграмотності у школах та університетах	Розвиток критичного мислення, збільшення рівня медіаграмотності суспільства
	Персональний: асистент для власного аналізу інформації	Розвиток критичного мислення
	Моніторинг ЗМІ: аналіз та класифікація джерел інформації вповноваженими органами як пропагандистські чи не пропагандистські	Автоматизація ручного процесу аналізу об’єктивності ЗМІ

Важливим кроком у маркетинговому аналізі стартапу є аналіз конкурентів. Першим ідентифікованим конкурентом є простий інструмент виявлення пропаганди від університету Масарика в Чехії [35]. Іншим конкурентом зі схожою функціональністю є Blackbird.AI, зокрема, їх інструмент Compass Check для перевірки тверджень [36]. За визначенням розробників, Compass Context забезпечує ясність щодо онлайн-твердження, посилання на статтю або підтриманого посту чи відео в соціальних мережах. Коли ви ставите Compass Context запитання або вставляєте будь-яке посилання, він обробляє дані в режимі реального часу з тисяч джерел, перевіряє твердження, аналізує результати за допомогою платформи наративної аналітики Blackbird.AI і генерує точну відповідь зі зносками та посиланнями на цитати. Ще одним непрямим конкурентом є проєкт VoxCheck – фактчекінговий проєкт від незалежної аналітичної платформи Вокс Україна, команда якої займається виявленням пропаганди, фейків та маніпуляцій російських джерел [37]. Потрібно розглянути цих конкурентів детальніше, вони представлені у таблиці 5.2.

Таблиця 5.2.Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко-економічні характеристики ідеї	Товари/концепції конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	Propaganda detection tool by Masaryk University	VoxCheck	Blackbird.AI			
1.	Зручність користувацького інтерфейсу	+	-	-	+	-	+	-
2.	Автоматичне	+	+	-	+	-	+	-

	опрацювання тексту							
3.	Класифікація технік пропаганди у тексті	+	+	-	-	-	+	-
4.	Ідентифікація пропагандистських фрагментів	+	+	+	-	-	+	-
5.	Мультимовність	+	-	-	+	-	-	+
6.	Швидкодія	+	+	-	+	-	+	-
7.	Можливість інтеграції в інші сервіси	+	-	-	-	-	-	+
8.	Можливість застосування в освітній сфері	+	+	+	-	-	+	-
9.	Фізична інтеракція через робота	+	-	-	-	-	-	+
10.	Точність передбачення	-	-	+	-	-	-	+
11.	Розширюваність	+	-	-	-	-	-	+
12.	Аналіз будь-якого тексту	+	-	-	+	-	-	+

За результатами аналізу конкурентів можна стверджувати, що мій продукт відповідає майже всім важливим вимогам. Інші конкуренти, на противагу, програють у певних характеристиках за рахунок або своєї дещо іншої спеціалізації (Blackbird.AI), або відсутності автоматизації (VoxCheck), або ж меншої

функціональності (Propaganda Tool). Характеристикою, де мій продукт не є сильним, це точність, виміряна метриками, однак для освітньої сфери вона є менш важливою, оскільки продукт покликаний звертати увагу на потенційні маніпуляції і допомагати у вивченні їх варіацій. Також жодне із автоматизованих рішень не пропонує гарантованої точності, окрім VoxCheck, які проводять мануальні дослідження. Тому будемо стверджувати, що мій продукт є достойним суперником на ринку.

5.2 Технологічний аудит ідеї проєкту

Важливим етапом оцінки потенціалу стартапу є проведення технологічного аудиту. Аналіз технологічної здійсненності мого проєкту розміщено у таблиці 5.3.

Таблиця 5.3 Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Веб-застосунок	Python, Streamlit, CSS, HTML, OpenAI API	Наявні	Доступні за оплату
2	Інтелектуаьний робот та серверне рішення	Raspberry Pi, NAO, Pepper, OpenAI API, Python	Наявні	Доступні за оплату
3	Десктопний застосунок	Python, CUDA, Tensorflow, Gemma, JAN	Наявні	Доступні
Обрана технологія реалізації ідеї проекту: технологія 1				

Доцільно обрати технологію веб-застосунку, внаслідок легкості розробки та малих грошових затрат, а також можливості потенційної інтеграції рішення в інші системи, зокрема роботизовані. Так, у подальшому з системою можна буде взаємодіяти через API.

5.3 Аналіз ринкових можливостей запуску стартап-проєкту

У таблиці 5.4 зображено загальну попередню характеристику потенційного ринку мого стартап-проєкту.

Таблиця 5.4 Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	10
2	Загальний обсяг продаж, грн/ум.од	10000
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу	Освітні установи мають право вимагати докази надійності рішення
5	Специфічні вимоги до стандартизації та сертифікації	Потенційно освітні стандарти
6	Середня норма рентабельності в галузі (або по ринку), %	60

Видно, що ринок має нахил до зростання через активні інформаційні кампанії та війни по всьому світу і значну залученість населення в інформаційний простір. Дещо важко оцінити загальний грошовий обсяг продажів і рентабельність ринку, оскільки в освітній сфері багато клієнтів можуть залежати від ситуативного фінансування.

Потрібно розглянути тепер характеристики потенційних сегментів клієнтів стартап-проєкту.

Таблиця 5. 5 Характеристика потенційних клієнтів стартап-проєкту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові	Відмінності у поведінці різних	Вимоги споживачів до товару

		сегменти ринку)	потенційних цільових груп клієнтів	
1	Розвиток медіаграмотності підростаючого покоління, щоб збільшити розумовий потенціал країни	Школи, державні установи, неприбуткові організації	Зацікавлені у підвищенні медіаграмотності серед учнів	Дружній користувацький інтерфейс, точність передбачення, внесення ігрового елементу
2	Потреба підтримувати власну медіагігієну і критичне мислення	Користувачі	Зацікавлені у застосуванні для власних потреб, можливо, бажають більше кастомізації	Точність, швидкість, більше функцій
3	Потреба інструменту для фактчекінгу та оцінки ЗМІ для формування їх рейтингу	Незалежні організації, фактчекінгові організації	Зацікавлені у об’єктивній оцінці публікацій видань	Точність, швидкість

Як видно, є декілька окремих напрямів клієнтського сегменту, які можна надалі опрацьовувати окремо. В кожному з них є перспектива запровадження рішення з врахуванням персональних потреб цієї групи.

Доцільно розглянути фактори загроз, які присутні для даного проєкту (таблиця 5.6).

Таблиця 5.6 Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренція	Багато конкурентів на ринку, наприклад, з дублюючим функціоналом або кращими показниками, оскільки інформаційна гігієна користується попитом	Покращення якості продукту та/або зменшення вартості
2	Фінансовий	При розробці додаткового функціоналу (наприклад, інтеграції з роботом) існують фінансові навантаження. Також OpenAI стягує плату за послуги	Проведення фінансового аналізу та розподілу кошторису, з плануванням бюджету наперед
3	Військові дії	Несе фізичну загрозу діяльності стартап-проєкту, а також його потенційним покупцям	Розробка Business Continuity Plan, плану безперервності бізнесу за непередбачених обставин

На кожну загрозу є розумний шлях вирішення, особливо ефективно планувати його наперед.

Розгляньмо фактори росту (можливостей) стартапу.

Таблиця 5.7 Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Створення нового продукту на основі наявного	Створення покращеного рішення (наприклад, повноцінного робота)	Періодичний розгляд можливості та ресурсів створення нового продукту

		для захоплення більшої кількості можливостей	
2	Оновлення технологій	Знаходження кращих технологій для реалізації функціоналу	Періодичний моніторинг нових технологічних розробок
3	Декларування інтелектуальної власності	Запровадження власної убезпеченості від крадіжки чи дублювання технологій	Патентування розробки

Оскільки присутні декілька факторів росту, потрібно їх розглянути та працювати в їх напрямку.

Необхідно розглянути тепер конкуренцію на ринку в ступеневому вимірі (таблиця 5.8).

Таблиця 5.8 Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Тип конкуренції олігополія	На ринку налічується декілька конкурентів зі схожими продуктами	Впровадження нових функцій, розумне співвідношення ціна/якість
2. Рівень конкурентної боротьби - глобальний	Продукт конкурує на міжнародному рівні	Пошук можливостей кастомного застосування в різних країнах та культурах
3. За галузевою ознакою - міжгалузева	Продукт може конкурувати в освітній сфері або сфері моніторингу ЗМІ	Пристосування продукту до різних галузей

4. Конкуренція за видами товарів: - товарно-видова	Конкуренція між схожими підходами до вирішення проблеми у різних продуктах	Покращення основного алгоритму, на якому працює рішення
5. За характером конкурентних переваг - цінова	Співвідношення ціна/якість	Введення розумної вартості на рішення з врахуванням тенденцій на ринку та собівартості
6. За інтенсивністю - марочна	Впізнаваність за брендами	Введення легко запам’ятовуваної назви, стратегія ведення бренду

Проведемо також аналіз конкуренції в галузі за М. Портером (таблиця 5.9).

Таблиця 5.9 Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Складові аналізу	Propaganda detection tool by Masaryk University, BlackbirdAI, VoxCheck	Можливі майбутні реалізації на основі LLM в зв’язку з їх популярністю	Вендори великих мовних моделей, зокрема OpenAI	Освітній ринок, ринок моніторингу ЗМІ	Немає
Висновки:	Певні конкуренти є доволі сильними, тому потрібно	Можливості входу на ринок присутні в найближчому	Постачальники LLM можуть диктувати умови, наприклад,	Ринок моніторингу ЗМІ може вимагати підвищенн	Немає

	постійно покращувати метрики власного рішення.	майбутньому із випущенням ефективніших моделей.	вартість використання їх моделей. Потрібно шукати шляхи оптимізації коштів.	я точності, освітній ринок – додавання нового функціоналу	

За аналізом конкуренції підсумуємо, що вихід на ринок можливий, ситуація з конкуренцією є не критичною.

Тепер обґрунтуємо фактори конкурентоспроможності продукту (таблиця 5.10).

Таблиця 5.10 Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування
1	Освітня цінність	Спрямованість на підвищення медіаграмотності – одна з ключових соціальних проблем сьогодення.
2	Простота інтеграції	Продукт легко інтегрується в існуючі освітні платформи або системи навчання.
3	Можливість масштабування	Продукт можна легко масштабувати, додавши підтримку нових мов або функціональність через оновлення програмного забезпечення. Також можливе розширення для різних цільових груп.

Описані чинники забезпечують стійку позицію продукту та сприяють його успішному впровадженню в умовах зростаючого попиту на медіаграмотність.

Потрібно тепер розглянути список сильних та слабких сторін продукту (таблиця 5.11).

Таблиця 5.11 Порівняльний аналіз сильних та слабких сторін

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з моїм продуктом						
			–3	–2	–1	0	+1	+2	+3
1	Користувацький інтерфейс	19		+					
2	Простота інтеграції	16	+						
3	Точність	10					+		
4	Швидкодія	12				+			
5	Вартість	17			+				

Завершимо етап аналізу ринкових можливостей складанням SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (таблиця 5.12) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін.

Таблиця 5.12 SWOT- аналіз стартап-проекту

Сильні сторони: Інноваційність: використання ШІ та робототехніки для інтерактивного навчання. Адаптивність: легко адаптується для різних застосувань. Зручність у використанні: інтуїтивний інтерфейс, інформаційна довідка, легка взаємодія для користувача.	Слабкі сторони: Залежність від зовнішніх технологій. Помилки в передбаченнях.
Можливості:	Загрози:

Можливість додати нові навчальні модулі або інші мови. Інтеграція з робототехнічними системами. Охоплення нових ринків.	Зростання кількості подібних освітніх ініціатив і технологічних рішень. Технологічні ризики: Швидкий розвиток технологій може зробити частину рішень застарілими.

Можливості для проекту переважають загрози, а сильні сторони – слабкі, отже, є перспективи йти далі.

Визначимо альтернативи ринкового впровадження стартап-проекту та проаналізуємо їхні ймовірності та терміни реалізації (табл. 5.13).

Таблиця 5.13 Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Бренд-менеджмент	Середня	1 рік
2	Маркетинг у соціальних мережах	Висока	2-4 місяці
3	Просування на освітніх та інших спеціалізованих подіях	Висока	3-6 місяців

Буде обрано просування продукту через маркетинг у соцмережах як альтернативний шлях впровадження, оскільки цей спосіб зараз є дуже популярним та ефективним через високу залученість людей у соцмережах і можливість налаштування якісного таргетування.

5.4 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку. Потрібно провести опис цільових груп потенційних споживачів у таблиці 5.14.

Таблиця 5.14. Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Приватні інноваційні школи	Висока	Середній	Низька	Висока
2	Державні освітні заклади	Середня	Середній	Низька	Низька
3	Журналістські організації	Низька	Низький	Низький	Середня
Які цільові групи обрано: приватні інноваційні школи.					

За результатами аналізу потенційних груп споживачів (сегментів) автори ідеї обирають цільові групи, для яких вони пропонуватимуть свій товар, та визначають стратегію охоплення ринку:

– якщо компанія зосереджується на одному сегменті – вона обирає стратегію концентрованого маркетингу;

– якщо працює із кількома сегментами, розробляючи для них окремо програми ринкового впливу – вона використовує стратегію диференційованого маркетингу;

– якщо компанія працює із всім ринком, пропонуючи стандартизовану програму (включно із характеристиками товару/послуги) – вона використовує масовий маркетинг.

Для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку продукту, її результат для даного проєкту наведено у таблиці 5.15.

Таблиця 5.15 Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Впровадження інтерактивного робота-асистента у навчальні заклади	Нішевий маркетинг (освітній сектор)	- Інноваційність: Поєднання ШІ, робототехніки та інтерактивного навчання. - Соціальна цінність: Відповідь на нагальні потреби суспільства щодо медіаграмотності.	Стратегія диференціації

Вибрана стратегія диференціації фокусується на створенні унікального продукту, який виділяється на ринку завдяки своїм інноваційним характеристикам і соціальній значущості. Нішевий підхід в освітньому секторі дозволяє точно адресувати потреби навчальних закладів, підвищуючи впізнаваність та ефективність впровадження.

Далі потрібно обрати стратегію конкурентної поведінки для даного продукту (таблиця 5.16).

Таблиця 5.16 Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідце	Чи буде компанія шукати нових	Чи буде компанія копіювати основні	Стратегія конкурентної

	м» на ринку?	споживачів, або забирати існуючих у конкурентів?	характеристики товару конкурента, і які?	поведінки*
	Ні, але є інноваційним гравцем	Основна мета – залучення нових споживачів, особливо серед освітніх установ	Ні, але враховуватиме кращі практики та шукатиме шляхи для покращення показників точності.	Стратегія зайняття конкурентної ніші

Проєкт не є «першопрохідцем», оскільки на ринку вже існують ініціативи з медіаграмотності. Однак унікальність рішення полягає в інтеграції ШІ з робототехнікою, що створює нову освітню нішу з більшою залученістю та інтерактивністю.

Головна стратегія полягає у формуванні нової аудиторії – навчальних закладів і державних освітніх програм, а не конкуруванні з існуючими проєктами. Це дозволяє зосередитися на співпраці з організаціями, які ще не мають подібних рішень.

Компанія не планує пряме копіювання характеристик конкурентів. Однак, для підвищення конкурентоспроможності будуть впроваджені кращі практики, такі як адаптивне навчання та інтерактивність, що є стандартними для провідних освітніх платформ.

Ця стратегія дозволяє зміцнити позиції проєкту в освітньому середовищі та створити власну унікальну пропозицію на ринку.

В таблиці 5.17 наведено визначення стратегії позиціонування для даного продукту.

Таблиця 5.17 Визначення стратегії позиціонування

№ п/	Вимоги до товару	Базова стратегія	Ключові конкурентоспромо	Вибір асоціацій, які мають сформувати

п	цільової аудиторії	розвитку	жні позиції власного стартап-проекту	комплексну позицію власного проекту (три ключових)
1	Інтерактивне навчання медіаграмотності	Розвиток через інноваційність і соціальну орієнтацію	- Інноваційність технологій: ШІ + робототехніка для навчання. - Висока освітня цінність: Розвиток медіаграмотності.	- Технологічна передовість: Використання новітніх рішень. - Соціальна відповідальність: Боротьба з пропагандою та фейками.

Ця стратегія створює чітке позиціонування, яке допоможе диференціювати продукт на ринку та залучити зацікавлені сторони.

5.5 Розроблення маркетингової програми стартап-проекту

Першим кроком є формування маркетингової концепції товару, який отримає споживач. Для цього у табл. 18 підсумуємо результати попереднього аналізу конкурентоспроможності товару.

Таблиця 18. Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити
1	Підвищення рівня медіаграмотності серед учнів і студентів	Ефективне навчання через технології	Інноваційний формат навчання: Поєднання ШІ та робототехніки.
2	Боротьба з дезінформацією	Точне виявлення пропагандистськи	Висока точність аналізу: Використання передових моделей

	та пропагандою	х технік у текстах	ШІ для розпізнавання пропаганди.

Проєкт відповідає ключовим потребам сучасної освітньої сфери, зокрема підвищенню медіаграмотності та боротьбі з пропагандою. Створення додаткових функцій, таких як ігрові елементи або багатомовна підтримка, дозволить проєкту залишатися конкурентоспроможним і розширювати цільову аудиторію на інші сегменти.

В таблиці 5.19 розміщено три рівні моделі даного товару.

Таблиця 5.19 Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Інтелектуальний робот-асистент для виявлення пропаганди		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Простота у користуванні	М	Е
	2. Достатня точність передбачення	М	Тл
	3. Вартість	М	Вр
	Якість: відповідна нормам програмного забезпечення		
	Пакування: немає		
	Марка: DefAIne		
III. Товар із підкріпленням	До продажу: Інтелектуальний робот-асистент з виявлення пропаганди		
	Після продажу: зв’язок з клієнтами, покращення і впровадження нових технологій у продукт		
За рахунок чого потенційний товар буде захищено від копіювання: права на інтелектуальну власність			

Визначимо тепер межі встановлення ціни на продукт (таблиця 5.20).

Таблиця 5.20 Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	100-200$	50-100$	5000$	10-10$

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (таблиця 5.21):

Таблиця 5.21 Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
	Потрібно придбати програмне забезпечення на сайті	Вебсайт	Нульовий	Вебсайт

Отже, основною системою збуту буде продаж через вебсайт.

Таблиця 5.22 Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Прагнення до інноваційних методів	Освітні форуми, конференції,	- Інноваційність продукту	Сформувати інтерес до нового	Технологічний прорив у боротьбі з

	навчання, увага до соціальної відповідальності	соціальні мережі, вебінари, тематичні ЗМІ	- Соціальна значущість - Освітня ефективність	інтерактивного навчального продукту, підвищити обізнаність про медіаграмотність	дезінформацією та навчанні критичному мисленню.

Цільові клієнти (освітні установи, державні органи, громадські організації) демонструють високий інтерес до інноваційних підходів у навчанні. Вони прагнуть підвищити рівень медіаграмотності серед учнів, використовуючи сучасні технології та інтерактивні методи.

Основними каналами для залучення цільової аудиторії є професійні освітні конференції, соціальні медіа (LinkedIn, YouTube), тематичні вебінари та спеціалізовані ЗМІ. Ці платформи дозволяють максимально ефективно доносити інформацію про інноваційний продукт, особливо із застосуванням таргетованої реклами та маркетингової стратегії.

Позиціонування базується на трьох головних елементах: інноваційності, соціальній значущості та освітній ефективності. Завдання рекламного повідомлення полягає у підвищенні обізнаності про медіаграмотність, а також демонстрації того, як робот-асистент допомагає у боротьбі з дезінформацією.

5.6 Висновки

У п’ятому розділі було проведено маркетингове дослідження стартап-проєкту інтелектуального робота-асистента для виявлення пропаганди у новинних текстах. Було підтверджено потенціал комерціалізації продукту.

Детально було розглянуто потенційні ніші для входу на ринок, клієнтські сегменти (освітній ринок у вигляді шкіл та університетів, ЗМІ для фактчекінгу і

перевірки об’єктивності, фізичні особи для персонального використання). Також було проведено аналіз прямих і непрямих конкурентів, і виявлено, що їх наявність не обмежує можливість виходу даного продукту на ринок, а навпаки, спонукає його.

Основною загрозою впровадження стартапу є залежність від певних технологій і неідеальна точність. Частково неідеальною точністю можна пожертвувати у освітньому застосуванні, оскільки ціль інтелектуального помічника – вказати на місця в тексті, в яких слід бути уважними. Також можна в перспективі подальших досліджень цю точність удосконалити. Водночас залежності від технологій неможливо повністю уникнути, але варто враховувати нюанси їх використання.

Загалом за результатами проведеного аналізу можна стверджувати, що впровадження розробки робота-асистента з виявлення пропаганди в життя є перспективним проєктом.

ВИСНОВКИ

В результаті виконання роботи було розроблено інтелектуального робота-асистента на основі великих мовних моделей для виявлення пропаганди. В процесі виконання було проведено огляд існуючих рішень, і визначено аспекти, які ще не піддавалися дослідженню, наприклад, використання двомовного датасету, і доналаштування великих мовних моделей сімейства Gemma/Gemma 2. Було зібрано набір даних із пропагандистськими техніками та фрагментами тексту, де вони зустрічаються, і проведено точне налаштування моделей GPT-4o mini та Gemma/Gemma 2 для завдання класифікації технік та ідентифікації пропагандистських фрагментів. Порівняно отримані метрики для різних конфігурацій доналаштованих моделей. Можна зробити висновок, що модель GPT -4o mini краще впоралась із поставленим завданням, ніж моделі Gemma/Gemma 2, а також, що точне налаштування дає вищі значення метрик, ніж базова модель. Також отримані моделі конкурують з отриманими в попередніх дослідженнях на цю тему, маючи вищі, або ж рівні, зате з використанням меншої та більш економічно ефективної моделі показники.

Далі було використано найкращу за показниками конфігурацію моделі у розробці інтелектуального асистента. Розроблений продукт представляє собою веб-застосунок з функцією аналізу вхідного тексту на пропаганду. Демонструючи виявлені техніки пропаганди та відповідні фрагменти у тексті, а також інформуючи користувача про існуючі техніки з поясненнями та прикладами, він може бути застосований у освітній сфері при навчанні медіаграмотності. Було продемонстровано роботу інтелектуального асистента на реальних екземплярах текстів з пропагандистських джерел. Асистент показав свою здатність як ідентифікувати присутність технік пропаганди, так і видавати відповідні негативні результати при їх відсутності.

Результат розробки можна інтегрувати з іншими системами і використовувати як повноцінного фізичного робота під час навчання за допомогою певних технологій, описаних в роботі.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Пропаганда – Вікіпедія. URL: https://uk.wikipedia.org/wiki/Пропаганда (дата звернення: 24.10.2024).

2. Aslanova V. Psychological support assistant based on fine-tuned LLaMA 3 model / Aslanova V., Oliinyk V. // The International Conference on Security, Fault Tolerance, Intelligence ICSFTI2024 (June 07, 2024, Kyiv, Ukraine), page 1-13. https://icsfti-proc.kpi.ua/article/view/309532 (дата звернення: 05.12.2024).

3. Oliinyk V. Data augmentation with foreign language content in text classification using machine learning / Oliinyk V., Osadcha K. // Adaptive systems of automatic control, 2020. Vol. 1, №36. – P. 51-59.

4. Advanced NLP Techniques for Text Classification. AST Consulting. URL: https://astconsulting.in/blog/2023/07/10/advanced-nlp-techniques-text-classification/ (дата звернення: 24.10.2024).

5. Papay, Sean, Roman Klinger, and Sebastian Padó. Dissecting span identification tasks with performance prediction. arXiv preprint arXiv:2010.02587, 2020.

6. Large language model – Wikipedia. Wikipedia, the free encyclopedia. URL: https://en.wikipedia.org/wiki/Large_language_model (дата звернення: 24.10.2024).

7. Da San Martino, G., Seunghak, Y., Barrón-Cedeno, A., Petrov, R., & Nakov, P. (2019). Fine-grained analysis of propaganda in news article. In Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP) (pp. 5636-5646). Association for Computational Linguistics.

8. Gupta, P., Saxena, K., Yaseen, U., Runkler, T., & Schütze, H. "Neural architectures for fine-grained propaganda detection in news." arXiv preprint arXiv:1909.06162, 2019.

9. Sprenkamp, Kilian, Daniel Gordon Jones, Liudmila Zavolokina. "Large language models for propaganda detection." arXiv preprint arXiv:2310.06422, 2023.

10. Vijayaraghavan, Prashanth, and Soroush Vosoughi. "TWEETSPIN: Fine-grained propaganda detection in social media using multi-view representations."

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2022.

11. Dimitrov, Dimitar, et al. "Detecting propaganda techniques in memes." arXiv preprint arXiv:2109.08013, 2021.

12. What is a Large Language Model (LLM) - GeeksforGeeks. GeeksforGeeks. URL: https://www.geeksforgeeks.org/large-language-model-llm/ (дата звернення: 24.10.2024).

13. Поночовний П.С. Аналітичний огляд способів застосування великих мовних моделей (LLM) для вирішення прикладних задач / Поночовний П.С., Олійник В.В.// Інженерія програмного забезпечення і передові інформаційні технології (Soft Tech-2023): матеріали V Міжнародної науково-практичної конференції молодих вчених та студентів, 19-21 грудня 2023 року, м. Київ, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», 2023. С. 272-276. URL: https://drive.google.com/file/d/1racc22TBKkFFNzBRSzOrePKpFfbnGDJ1/view (дата звернення: 02.12.2024).

14. Number of ChatGPT Users and Key Stats (October 2024). NamePepper. URL: https://www.namepepper.com/chatgpt-users (дата звернення: 24.10.2024).

15. GPT-4o mini: advancing cost-efficient intelligence. OpenAI. URL: https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/ (дата звернення: 24.10.2024).

16. Ashish Vaswani, Noam Shazeer, Niki Parmar. Attention Is All You Need. URL: https://arxiv.org/pdf/1706.03762v7 (дата звернення: 24.10.2024).

17. Gemma: Open Models Based on Gemini. Research and Technology. URL: https://storage.googleapis.com/deepmind-media/gemma/gemma-report.pdf (дата звернення: 24.10.2024).

18. What is instruction tuning? URL: https://www.ibm.com/topics/instruction-tuning (дата звернення: 24.10.2024).

19. Тонке настроювання (глибоке навчання) – Вікіпедія URL: https://uk.wikipedia.org/wiki/%D0%A2%D0%BE%D0%BD%D0%BA%D0%B5_%D0

%BD%D0%B0%D1%81%D1%82%D1%80%D0%BE%D1%8E%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_(%D0%B3%D0%BB%D0%B8%D0%B1%D0%BE%D0%BA%D0%B5_%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F) (дата звернення: 24.10.2024).

20. Oliinyk V. An efficient face mask detection model for real-time applications / Oliinyk V., Ryzhiy A. // Adaptive systems of automatic control, 2022. Vol. 1, №40. – P. 54-64.

21. Oliinyk V. Low-resource text classification using cross-lingual models for bullying detection in the Ukrainian language / Oliinyk V., Matviichuk I. // Adaptive systems of automatic control, 2023. Vol. 1, №42. – P. 87-100.

22. Fine-tuning. OpenAI Platform Docs. URL: https://platform.openai.com/docs/guides/fine-tuning (дата звернення: 24.10.2024).

23. Disinformation Detection Challenge by AI HOUSE x Mantis Analytics. Kaggle: Your Machine Learning and Data Science Community. URL: https://www.kaggle.com/competitions/disinformation-detection-challenge/data (дата звернення: 24.10.2024).

24. Preparing Your Dataset: Fine-tuning. OpenAI Platform. URL: https://platform.openai.com/docs/guides/fine-tuning/preparing-your-dataset (дата звернення: 24.10.2024).

25. Тензорний блок обробки – Вікіпедія. URL: https://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BD%D0%B7%D0%BE%D1%80%D0%BD%D0%B8%D0%B9_%D0%B1%D0%BB%D0%BE%D0%BA_%D0%BE%D0%B1%D1%80%D0%BE%D0%B1%D0%BA%D0%B8 (дата звернення: 24.10.2024).

26. Variants. Low-rank adaptation – Wikipedia. URL: https://en.wikipedia.org/wiki/Fine-tuning_(deep_learning)#Low-rank_adaptation (дата звернення: 24.10.2024).

27. Qianli Zhu. Distributed training with Keras 3. URL: https://keras.io/guides/distribution/ (дата звернення: 24.10.2024).

28. difflib – Helpers for computing deltas. URL: https://docs.python.org/3/library/difflib.html (дата звернення: 24.10.2024).

29. Pepper (Robot) – Wikipedia. URL: https://en.wikipedia.org/wiki/Pepper_(robot) (дата звернення: 24.10.2024).

30. mBot Robot Kit – Makeblock. URL: https://www.makeblock.com/pages/mbot-robot-kit (дата звернення: 24.10.2024).

31. Raspberry PI Software. – Wikipedia URL: https://en.wikipedia.org/wiki/Raspberry_Pi#Software (дата звернення: 24.10.2024).

32. Speech Synthesis – Wikipedia. URL: https://en.wikipedia.org/wiki/Speech_synthesis#Text-to-speech_systems (дата звернення: 24.10.2024).

33. Why LLMs Can't Count the R's in 'Strawberry' & What It Teaches Us. URL: https://arbisoft.com/blogs/why-ll-ms-can-t-count-the-r-s-in-strawberry-and-what-it-teaches-us#the-case-of-strawberry (дата звернення: 24.10.2024).

34. Іноземні голоси російської пропаганди. Укрінформ. URL:https://www.ukrinform.ua/rubric-polytics/3670440-inozemni-golosi-rosijskoi-propagandi.html (дата звернення: 24.10.2024).

35. Ondřej, H. Propaganda detection tool. URL: https://www.muni.cz/en/research/publications/1631559 (дата звернення: 24.10.2024).

36. Compass Context. Verify and contextualize the information you see online. URL:https://blackbird.ai/compass-context/ (дата звернення: 24.10.2024).

37. Вокс Україна. VoxCheck. URL: https://voxukraine.org/voxcheck (дата звернення: 24.10.2024).