Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)
Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів виявлення дезінформації в джерелах інформації з використанням штучних нейронних мереж
(тема)

Виконав:
здобувач другого року навчання,
групи ДСм-24-1

Сербін О.В
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки

(код і повна назва спеціальності)
Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма Науки про дані (Data Science)

(повна назва спеціалізації)
Керівник проф. Філатов В.О.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Лариса ЧАЛА
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Науки про дані (Data Science)
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20 __ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Сербіну Олексію Вячеславовичу
(прізвище, ім'я, по батькові)

1. Тема роботи: Дослідження методів виявлення дезінформації в джерелах інформації з використанням штучних нейронних мереж

затверджена наказом по університету від "24" листопада 2025 р. № 1057Ст

2. Термін здачі студентом роботи до екзаменаційної комісії 17 грудня 20 25 р.

3. Вихідні дані до роботи Науково-технічні публікації, Python documentation, набори даних для тренування та тестування системи, попередньо навчені моделі, локально розвернута велика мовна модель

4. Перелік питань, що потрібно опрацювати в роботі

1) Огляд проблеми дезінформації та методів її виявлення

2) Огляд датасетів та підходів на основі нейронних мереж для задачі виявлення дезінформації в текстах

3) Застосування підходів та аналіз їх результатів

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	24.11.2025	виконано
2	Аналіз та розбір типів дезінформації	26.11.2025	виконано
3	Розбір традиційних методів боротьби з	28.11.2025	виконано
	дезінформацією		
4	Пошук датасетів та їх попередній аналіз даних	30.11.2025	виконано
5	Огляд рекурентних нейронних мереж	01.12.2025	виконано
6	Огляд трансформерів та аналіз підходів в fine-	02.12.2025	виконано
	tuning		
7	Пошук предтренованих моделей для виконання	04.12.2025	виконано
	задач		
8	Огляд великих мовних моделей та RAG підходу	05.12.2025	виконано
9	Аналіз результатів моделей на основі	07.12.2025	виконано
	рекурентних нейронних мереж		
10	Аналіз результатів моделей на основі	09.12.2025	виконано
	трансформерів		
11	Створення робочих процесів застосування	10.12.2025	виконано
	великої мовної моделі		
12	Аналіз результатів підходів на основі RAG	12.12.2025	виконано
13	Оформлення пояснювальної записки	15.12.2025	виконано
14	Захист перед ЕК	17.12.2025	виконано

Дата видачі завдання 24 листопада 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ проф. Філатов В.О.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 62 с., 12 рис., 22 табл., 1 дод., 16 джерел.

ВЕКТОРНІ БАЗИ ДАНИХ, ДАТАСЕТ, ДЕЗІНФОРМАЦІЯ, ІНТЕЛЕКТУАЛЬНА СИСТЕМА, МАНІПУЛЯЦІЯ, МЕХАНІЗМ УВАГИ, НАРАТИВ, НЕЙРОННІ МЕРЕЖІ, РЕКУРЕНТНІ НЕЙРОННІ МЕРЕЖІ, ТРАНСФОРМЕРИ, ФАКТЧЕКІНГ, ФЕЙК, FINE-TUNING, LLM, RAG.

Об’єктом дослідження роботи є процес виявлення та класифікації дезінформації у текстових даних.

Предмет дослідження: підходи та методи, які можна використовувати для створення інтелектуальної системи для розпізнавання дезінформації, яка представлена в текстовому вигляді.

Мета роботи: оцінити ефективність різних підходів та методів нейронних мереж на задачі виявлення дезінформації.

Методи дослідження: методи обробки природньої мови, побудови та навчання нейронних мереж, оцінки метрик якості роботи моделі.

Під час виконання роботи було визначено, що саме з себе представляє дезінформація та який вигляд вона може мати. Далі було розглянуто її вплив на суспільство в наші дні та за допомогою яких джерел вона поширюється. Був проведений огляд традиційних методів боротьби з дезінформацією. Далі були запропоновані та детально проаналізовані підходи, які використовують нейронні мережі, що можуть бути використані для розробки системи, яка зможе з нею боротися. Вирішення проблеми дезінформації було розглянуто у задачах виявлення неправдивих тверджень та маніпуляції в текстах.

ABSTRACT

Master’s thesis contains: 62 pp., 12 fig., 22 tabl., 1 ann., 16 references.

ATTENTION MECHANISM, DATASET, DISINFORMATION, FACT-CHECKING, FAKE, FINE-TUNING, INTELLIGENT SYSTEM, LLM, MANIPULATION, NARRATIVE, NEURAL NETWORKS, RAG, RECURRENT NEURAL NETWORKS, TRANSFORMERS, VECTOR DATABASES.

The object of the study is the process of detecting and classifying disinformation in textual data.

The subject of the study: approaches and methods that can be used to create an intelligent system for recognizing disinformation presented in textual form.

Research goal: to evaluate the effectiveness of various neural network approaches and methods for the task of disinformation detection.

Research methods: natural language processing methods, construction and training of neural networks, and evaluation by model quality metrics.

During the work, the nature of disinformation and the forms it can take were defined. The study then examined its impact on society today and the channels through which it is spread. A review of traditional methods for combating disinformation was conducted. Next, approaches that use neural networks and could be employed to develop systems that fight disinformation were proposed and analyzed in detail. The problem of disinformation was considered in tasks of detecting false claims and manipulation in texts.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

РНМ – рекурентна нейронна мережа;

GRU – Gated Recurrent Unit – блокова рекурентна одиниця;

LSTM – Long Short-Term Memory – довгострокова короткочасна пам'ять;

LLM – Large Language Model – велика мовна модель;

RAG – Retrieval-Augmented Generation – генерація з розширеним пошуком;

RNN – Recurrent Neural Network – рекурентна нейронна мережа.

ВСТУП

У сучасному інформаційному середовищі, яке характеризується швидким поширенням текстового контенту через соціальні мережі, месенджери та цифрові медіаплатформи, завдання автоматичного виявлення недостовірної або маніпулятивної інформації набуває критичного значення для збереження якості дискусій та інформаційної безпеки суспільства. Разом із зростаючою кількістю даних зростають і складність та різноманіття прийомів, що використовуються для масового поширення хибних та небезпечних наративів. Наприклад, покращуються стилістичні варіації, контекстні підміни, застосовуються мережі ботів та гібридні форми контенту, які поєднують правду з вигадкою. Це створює вимогу до розробки методів, здатних не лише враховувати лінгвістичні ознаки текстів, але й знаходити більш глибокі семантичні закономірності, а також адаптуватися до змін у поведінці джерел інформації й способах маніпуляцій.

У сфері обробки текстової інформації сьогодні існує велика кількість нейромережевих архітектур та підходів, кожен з яких має свої сильні й слабкі сторони. Доступні підходи глибинного навчання, зокрема рекурентні мережі та трансформери дають змогу автоматично виділяти важливі ознаки тексту, розуміти контекст і приховані смислові зв’язки. Вони здатні працювати з великим об’ємом даних і вчитися на реальних прикладах, що робить їх об’єктом уваги для задач виявлення недостовірних повідомлень. Проте для ефективного застосування цих підходів необхідно детально розглянути їхню роботу, можливості та обмеження, а також протестувати та проаналізувати їх результати на різних датасетах.

1 ОГЛЯД ПРОБЛЕМИ ДЕЗІНФОРМАЦІЇ ТА МЕТОДІВ ЇЇ ВИЯВЛЕННЯ

1.1 Проблема дезінформації в сучасному інформаційному просторі

Сьогодні для опису інформаційних порушень використовується різна термінологія, що дозволяє розмежувати їх за наміром. Виділяють три основні категорії:

– місінформація – неумисне поширення неточних або хибних повідомлень [1]. У цьому випадку відсутній злий намір, а саме поширення відбувається через помилку, недбалість або неправильне розуміння фактів;

– дезінформація – це умисно створена та поширена хибна інформація, головною метою якої є обман, маніпуляція чи завдання шкоди;

– малінформація – це навмисна публікація приватної інформації в особистих, корпоративних чи політичних, а не суспільних інтересах. Це також може включати навмисну зміну контексту, дати чи часу оригінального контенту [1].

У цій роботі основна увага буде приділена саме дезінформації. Її ключовою метою є введення в оману цільової аудиторії, маніпуляція громадською думкою, отримання політичних чи матеріальних переваг, або підрив довіри до суспільних та державних інституцій. Дезінформація може бути представлена у різних форматах: візуальному, аудіовізуальному та текстовому. У текстах дезінформація може виглядати особливо переконливо. Це тому, що їх легко зробити схожими на авторитетні джерела, складні ідеї можна звести до емоційних заголовків, а важливий контекст або цитати легко спотворити чи прибрати. Прояви текстової дезінформації можуть бути різними. Найчастіше це:

– повністю вигадані історії, які подаються як справжні новини;

– перекручені або вирвані з контексту цитати, що змінюють зміст сказаного;

– посилання на неіснуючих або непрофесійних «експертів»;

– використання слів і формулювань, які навмисно впливають на емоції або вводять в оману.

Окрім прямих фальсифікацій, використовуються більш приховані методи. До них належить: ігнорування або зміщення часових рамок подій, змішування достовірних фактів з вигадкою у пропорції, що створює ілюзію правдоподібності, а також подання статистичних даних без вказання джерел або методології їх збору.

Для глибокого аналізу дезінформації недостатньо зафіксувати лише її наявність. Також, необхідно класифікувати конкретні форми, яких вона набуває. Можна виділити наступні основні форми дезінформації, що проявляються у текстовому контенті.

Повністю вигаданий контент [2]. Представляє з себе найбільш чисту форму дезінформації, що полягає у створенні тексту, який не має жодної фактичної основи. Незважаючи на те, що такі матеріали повністю вигадані, вони подаються аудиторії як достовірні факти:

– маніпульований контент [2]. Цю форму створюють, беручи за основу справжню інформацію і навмисно її змінюючи. Маніпуляція може полягати у вириванні фрази з контексту, додаванні неправдивих слів або спотворенні даних, щоб змінити початковий зміст;

– контент-імітатор [2]**.** Створення тексту, що мімікрує під авторитетне джерело. Дезінформація подається таким чином, ніби її автором є відоме ЗМІ, поважний експерт або офіційна установа. Це досягається шляхом копіювання стилю, логотипів або прямої фальсифікації авторства;

– хибний контекст [2]. Подання достовірної інформації у навмисно спотвореному контексті. Найчастіше це маніпуляції з часовими рамками, тобто стара новина подається як актуальна, географічною прив'язкою або суб'єктами події, що кардинально змінює сприйняття факту;

– контент, що вводить в оману [2]. Це форма маніпуляції, коли текст формально не містить прямої брехні, але побудований так, щоб

підштовхнути читача до неправильного висновку. Це роблять, вибірково подаючи факти, приховуючи важливу інформацію або маніпулюючи порядком, у якому подані тези;

– хибний зв'язок [2]. Використання заголовків, зображень або підписів, які не відповідають суті основного тексту. Емоційний або провокативний заголовок привертає увагу, однак сам текст або не містить обіцяної інформації, або суперечить їй;

– сатира або пародія [2]. Це контент, спочатку створений для гумору, але згодом поширюється без цього пояснення. Аудиторія може сприйняти сатиричний текст як серйозне повідомлення і далі поширювати його як «реальну» інформацію.

Проблема дезінформації на сучасному етапі вийшла за межі поодиноких фейків і перетворилася на системну багатовимірну загрозу для глобальної стабільності, яка зачіпає здоров’я населення, суспільну довіру до ключових інституцій і стійкість демократичних процесів. Яскравим прикладом такого системного впливу стала криза COVID-19, під час якої Всесвітня організація охорони здоров’я ввела поняття «інфодемія» – ситуація, в якій переважають хибні або маніпулятивні твердження, що призводить до плутанини в сприйнятті критично важливих повідомлень, до ризикової поведінки й ускладнює боротьбу з хворобами, а також підриває довіру до органів охорони здоров’я й експертних рекомендацій [3]. Крім того, сьогодні організовані кампанії дезінформації регулярно фіксують як ключовий фактор, що формує громадську думку. Координація публікацій, мережі фальшивих профілів і програми для поширення створюють ілюзію широкої підтримки фальшивих думок та наративів, підсилюючи політичну поляризацію й збільшуючи ризики для національної безпеки. В Україні ця загроза має особливу інтенсивність через триваючу інформаційну та повномасштабну війну. Цілеспрямовані кампанії намагаються підривати довіру до державних інституцій, армії й міжнародних партнерів, підмінювати контексти та штучно легітимізувати псевдоекспертні позиції.

У сучасному інформаційному середовищі дезінформація поширюється надзвичайно швидко. Її джерела надзвичайно різноманітні і керуються різними мотивами. До основних належать:

– уряди та пов'язані з ними організації – використовують дезінформацію як інструмент гібридної війни для дестабілізації інших країн, впливу на вибори, посіювання розбрату та просування власних геополітичних інтересів;

– політичні групи – можуть використовувати дезінформацію для дискредитації опонентів, мобілізації свого електорату за допомогою емоційних та неправдивих гасел або для просування своїх інтересів;

– економічно мотивовані суб'єкти – створюють фейкові новини та клікбейтний контент виключно заради фінансової вигоди, отримуючи дохід від реклами за рахунок великої кількості переглядів;

– окремі люди зі своїми цілями – можуть створювати дезінформацію заради розваги, особистої помсти, привернення уваги або через щиру віру в певні теорії змови.

Канали поширення дезінформації є не менш різноманітними, ніж її джерела, і сучасні технології значно прискорюють цей процес. Центральну роль сьогодні відіграють соціальні мережі такі як Facebook, X, TikTok, Telegram, тощо. Їхні алгоритми, які оптимізовані для максимального залучення користувачів, часто сприяють вірусному поширенню саме емоційного, провокаційного та шокуючого контенту, яким і може бути дезінформація. Соцмережі дозволяють створювати так називаємі "ехо-камери", де користувачі бачать лише ту інформацію, що підтверджує їхні погляди, роблячи їх більш вразливими до маніпуляцій [4]. Важливим каналом є сайти фейкових новин, які імітують вигляд легітимних ЗМІ, але публікують вигадані історії. Вони поширюються через ті ж соціальні мережі, набуваючи фальшивої легітимності.

1.2 Традиційні підходи до виявлення дезінформації

Фактчекінг та експертна перевірка є фундаментальними та найбільш надійними методами виявлення дезінформації, особливо у випадку текстового контенту, де маніпуляції часто є прихованими або витонченими. Процес передбачає порівняння заяв, тверджень і статистичних даних із авторитетними джерелами, первинними документами, академічними публікаціями, офіційною статистикою, а також прямою перевіркою цитат спікерів на предмет викривлення контексту. Експертна перевірка слугує доповненням до цього процесу, залучаючи фахівців із конкретних галузей знань. Ці експерти мають можливість оцінити логічну коректність, точність використаної термінології, обґрунтованість висновків та ідентифікувати складні маніпуляції, які не є очевидними для звичайної людини.

Ще одним доповненням до фактчекінгу є лінгвістичний аналіз, оскільки він дозволяє виявляти приховані маніпуляції, вбудовані у стиль і структуру самого тексту. На відміну від перевірки фактів, цей підхід зосереджується на тому, як інформація подається. У дезінформаційних текстах шукаються специфічні маркери маніпуляції. До них належить використання дуже емоційної лексики, наприклад слів, що викликають страх, гнів або образу, застосування риторичних прийомів, як запитань чи перебільшень, які замінюють логічні докази, а також представлення оціночних суджень як доведених фактів. Ключовим елементом є аналіз того, як автор будує свою історію. Мається на увазі чи є він послідовним, і чи не використовує він вибіркове висвітлення, приховуючи важливі деталі, що змінили б сприйняття події. Лінгвістичний аналіз також оцінює стиль тексту, визначаючи, чи він намагається імітувати офіційний або експертний тон для підвищення довіри.

Попри свою надійність класичні методи виявлення дезінформації, мають низку суттєвих обмежень. Головна проблема полягає у масштабованості та часі. Людський фактчекінг є доволі повільним

процесом, бо вимагає залучення кваліфікованих експертів і журналістів для перевірки кожного повідомлення. Лінгвістичний аналіз також вимагає глибокого ручного втручання і не здатний обробити гігантські обсяги текстового контенту, який генерується щодня. Ці методи ефективні для перевірки резонансних і важливих історій, але виявляються недостатніми для протидії масовим, координованим кампаніям, що ширяться у соціальних мережах та месенджерах дуже швидко. Ситуація ускладнюється великим впливом штучного інтелекту на процес створення дезінформації. Новітні генеративні моделі дозволяють створювати правдоподібний та стилістично різноманітний текстовий контент, включаючи сотні варіацій одного фейкового наративу, швидко адаптувати його під різні аудиторії та платформи. Це значно знижує вартість створення дезінформації та підвищує складність її виявлення.

Але у той же час, штучний інтелект можна використовувати для боротьби з дезінформацією. Він може виявляти стилістичні аномалії, шукати відповідні джерела, класифікувати правдиві та неправдиві тексти. Тому мета цієї роботи: перевірити різні підходи глибинного навчання, як старі так і сучасні, щоб зрозуміти, які з них краще працюють у цих задачах.

2 ОГЛЯД ДАТАСЕТІВ ТА ПІДХОДІВ НА ОСНОВІ НЕЙРОННИХ МЕРЕЖ ДЛЯ ЗАДАЧІ ВИЯВЛЕННЯ ДЕЗІНФОРМАЦІЇ В ТЕКСТАХ

2.1 Постановка задачі та вимоги до моделей

У цій роботі задача виявлення дезінформації розглядається на прикладі визначення чи є текстова інформація правдивою або неправдивою. Для успішного виконання завдання від моделі вимагається розрізняти правдиві та хибні тексти, причому хибні можуть варіюватися від цілком нейтральних до тих, що містять явні ознаки маніпуляції або якісь стилістичні конструкції, які їх виділяють. Також можуть траплятися окремі твердження, щодо яких модель має прямо визначити, чи вони правдиві або хибні, або чесно визнати, що доступної інформації недостатньо. Іноді формулювання бувають двозначними або висновок залежить від додаткових фактів. Саме у таких випадках модель повинна вказати на невизначеність. Сам же процес виглядає так: на вході маємо текст, наприклад новину, пост, статтю або іншу інформацію, а на виході система повертає результат відповідно до її основної задачі. Найпростішим варіантом виходу є бінарна класифікація, коли модель просто визначає одне з двох значень. У випадку задачі виявлення фейків це «правда» або «неправда». Більш розвинена система, яка використовує сучасні підходи та архітектури нейронних мереж, може не лише визначати недостовірність інформації, а й пояснювати свій висновок. Наприклад, вона може підсвічувати сумнівні фрагменти тексту або пропонувати посилання на перевірені джерела, або просто повернути аргументацію стосовно прийнятого рішення. Такий підхід робить модель більш корисною та прозорою для користувача. У ще ширшому варіанті система може мати доступ до актуальних даних і автоматично шукати підтвердження чи спростування тверджень у режимі реального часу. Таким чином, в цьому видаку задача вже формулюється як поєднання класифікації тексту і, при потребі, пояснення результату в зрозумілій для людини формі.

Критерії оцінки ефективності можуть відрізнятися залежно від того, яким є кінцевий результат роботи системи. У данній роботі основними для визначених задач метриками є класичні accuracy, recall, precision та F1-міра, оскільки вони необхідні, коли система працює у форматі класифікації. Розглянемо трохи детальніше представлені метрики:

– accuracy – це загальна точність, показує відношення кількості правильно класифікованих текстів до загального обсягу вибірки. Може бути необ’єктивною у випадку дисбалансу класів;

– precision – визначає, яка частка текстів, ідентифікованих моделлю як, наприклад, «фейк», насправді є такими. Тобто, якщо клас має високий precision, то можна довіряти моделі, коли вона визначає текст цим класом;

– recall – визначає, яка частка об'єктів, що насправді належать до певного класу, була успішно виявлена моделлю. Якщо клас має високий recall, це означає, що система бачить майже всі його випадки і ризик пропустити записи цього класу є мінімальними;

– F1-міра – визначає гармонійне поєднання точності та повноти, показуючи наскільки ефективно модель балансує між цими показниками. Якщо модель має високий F1-міру, це свідчить про її стабільність, бо вона не лише знаходить більшість фейків, але й робить це коректно, не плутаючи їх з правдивою інформацією.

2.2 Підготовка датасетів та попередня обробка даних

Для навчання моделей було використано кілька датасетів, орієнтованих на задачі виявлення фейкових новин та перевірки правдивості тверджень. Вони охоплюють різні формати даних, починаючи від новинних повідомлень закінчуючи короткими фактологічними заявам, що дозволяє оцінити роботу моделей у різних сценаріях детекції дезінформації. Нижче представлений список цих наборів даних:

– LIAR – це набір з коротких тверджень, де кожне має одну з шести міток правдивості, а саме: pants-fire, false, barely-true, half-true, mostly-true, true. Також присутні додаткова інформація у вигладі автора, партії, ресурсу, що дозволяє досліджувати не лише текстову, а й контекстну інформацію для покращення якості моделі;

– Politifact Fact-Check – датасет фактчекінгових тверджень, зібраних із сайту PolitiFact, одного з провідних інтернет-ресурсів для перевірки правдивості політичних висловлювань і заяв. PolitiFact сам по собі не є датасетом, а радше джерелом для таких датасетів. Редактори PolitiFact оцінюють твердження політиків, пости у соцмережах, виступи й інші висловлювання, присвоюючи їм вердикти правдивості, які потім зберігають у структурованому вигляді для досліджень і навчання моделей машинного навчання. Схожа структура використовується в LIAR, але представлений PolitiFact-датасет є розширеними за обсягом та метаданими порівняно з класичним LIAR, що може бути кращим варіантом для моделі;

– FEVER – це великий датасет для задач автоматичного фактчекінгу, який створений для оцінювання здатності моделей перевіряти правдивість тверджень на основі інформації з Вікіпедії. У цьому датасеті кожне твердження супроводжується міткою правдивості, яка може бути supported, тобто підтверджене наявними даними, refuted, що означає спростоване, або not enough info, тобто недостатньо інформації для винесення рішення. Суть задачі полягає в тому, що система повинна знайти відповідні докази твердження у статтях Вікіпедії і, спираючись на них, визначити, чи підтверджують чи спростовують вони дане твердження, або ж цих даних недостатньо для прийняття остаточного висновку;

– Real and Fake News – містить новинні статті, позначені як справжні або фейкові. У ньому представлені заголовки з текстами новин, причому тексти мають доволі великий обсяг. Головна особливість цього набору даних полягає в тому, що правдиві та фейкові новини значно відрізняються за стилем написання. Справжні новини зазвичай походять із авторитетних

новинних агентств і мають офіційний стиль, тоді як фейкові новини часто беруться з блогів, неперевірених сайтів або сумнівних джерел, і їхній стиль менш формальний та більш сенсаційний. Такий розділ може бути корисним, щоб подивитися чи зможе модель визначати правду та фейки на основі стилістичних ознак тексту.

Кожний датасет містить різну кількість записів. У таблицях 2.1 та 2.2 наведена відповідна інформація. В першій показані датасети для задач виявлення неправдивих текстів. Хоча деякі з них, а саме LIAR та PolitiFact, мають детальне ранжування міток, в цій таблиці вони були агреговні до двох типів: правдиві та неправдиві. В другій таблиці показана загальна кількість даних датасету FEVER .

Таблиця 2.1 – Кількість кожної з міток в датасетах задач виявлення неправдивих текстів

Датасет	Кількість правдивих тверджень	Кількість неправдивих тверджень
LIAR	4507	8282
Real and Fake news	21417	23481
PolitiFact Fact-Check	5795	15357

Таблиця 2.2 – Кількість кожної з міток в датасеті FEVER

Датасет	Кількість правдивих тверджень	Кількість неправдивих тверджень	Кількість невизначених тверджень
FEVER	124133	56872	79246

Перед безпосередньою розробкою та тренуванням моделі, крім оцінки розміру датасету, необхідно проаналізувати типи даних, розглянути кожну ознаку, а також видалити або замінити пропущені значення. На завершальному етапі формується тренувальна та тестова вибірки, якщо це не було представлено заздалегідь. Цей процес називається EDA.

Також, потрібно провести токенізацію текстової ознаки. Токенізація – це процес розбиття тексту на менші одиниці з метою подальшої обробки комп’ютером [5].

Далі коротко наведено приклад попереднього аналізу на основі датасету LIAR, щоб зрозуміти як вони будуть проводитися для інших наборів. Крім того, для інших датасетів у наступному розділі можуть бути представлені лише ті операції, які є специфічними та відмінними для кожного з них та можуть вплинути на саму модель.

Першочергово перевіримо типи даних для кожної ознаки. Результат для вибраного датасету представлено на рисунку 2.1. Як можна побачити, окрім ознак та їх типу, маємо ще інформацію про кількість не пустих значень. Подальша обробка цих даних залежить від характеру пропусків та типу ознаки. Наприклад, пусті значення для «Speaker job» або «Context» можна замінити на «Unknown», що дозволяє зберегти записи без втрати інформації. Додатково слід зазначити, що для деяких числових або категоріальних ознак можуть застосовуватися інші методи обробки пропусків: заповнення середнім значенням, медіаною, найчастішим варіантом або ж видалення записів із пропущеними даними, якщо їх кількість невелика.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10240 entries, 0 to 10239
Data columns (total 14 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   ID                    10240 non-null  object
 1   Label                 10240 non-null  object
 2   Statement             10240 non-null  object
 3   Subject               10238 non-null  object
 4   Speaker               10238 non-null  object
 5   Speaker job           7342 non-null   object
 6   State info            8030 non-null   object
 7   Party                 10238 non-null  object
 8   Barely true counts    10238 non-null  float64
 9   False counts          10238 non-null  float64
 10  Half true counts      10238 non-null  float64
 11  Mostly true counts    10238 non-null  float64
 12  Pants on fire counts  10238 non-null  float64
 13  Context               10138 non-null  object
dtypes: float64(5), object(9)
```

Рисунок 2.1 – Приклад представлення інформації про ознаки в датасеті

Для кожної категоріальної змінної підраховано число унікальних значень. У цьому датасеті всі категоріальні ознаки мають надто багато

унікальних значень. При такій ситуації застосування Label Encoding або One-Hot Encoding є неефективним. Перший підхід буде штучно створювати порядок, якого в даних немає, а другий призведе до різкого збільшення розмірності, що суттєво ускладнить обчислення та негативно позначиться на продуктивності моделі.

Крім того, було проаналізовано розподіл цільової ознаки. Частоти її значень були показані на гістограмі, що представлена на рисунку 2.2. Дивлячись на кількість записів цього датасету та кількість міток, можна сказати, що наявність такої великої кількості класів може значно ускладнити побудову моделі на основі цього набору даних. Тому для цього датасету була зменшена кількість типів міток до двох.

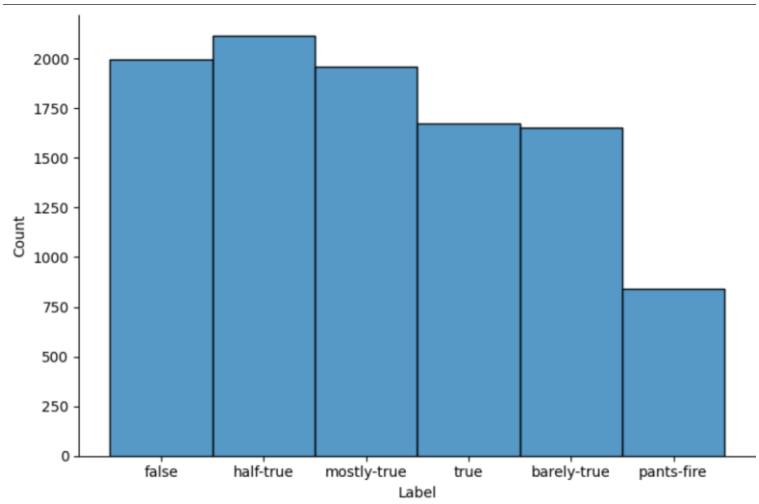

Рисунок 2.2 – Приклад гістограми частоти значень цільової мітки

Під час аналізу кожну ознаку оцінювали на предмет її корисності для навчання моделі. Як результат, для цього датасету, крім обов’язкових Label та Statement, було вирішено залишити Subject, Speaker, Speaker info та Context, які потенційно можуть мати позитивний вплив на навчання моделі, оскільки це дозволить їх опиратися не лише на основний текст, але й на додаткову інформацію, що потенційно підвищує точність передбачень та дозволяє глибше зрозуміти закономірності у даних.

Ознака Statement представляє безпосередньо текст, який модель має класифікувати як правдивий або фейковий. Як уже зазначалося, моделі не можуть працювати безпосередньо з текстом, тому його необхідно токенізувати. На сьогодні існує багато бібліотек для токенізації, кожна з яких використовує свої підходи:

– NLTK застосовує просте розділення тексту за пробілами та пунктуацією, а також регулярні вирази для виділення слів;

– spaCy використовує більш складні правила, враховуючи морфологію та контекст, що дозволяє коректно розпізнавати абревіатури чи складні словосполучення;

– токенізатори від Hugging Face пропонують підхід розбиття на підслова, використовуючи методи BPE або WordPiece, які розбивають слова на частини, що дозволяє обробляти невідомі слова.

Якісна токенізація зменшує кількість невідомих токенів і зберігає смислові частини слів, що допомагає моделі краще розуміти текст. Неправильна токенізація, навпаки, може створювати шум, розривати слова на недоцільні частини та знижувати точність моделі. Також важливо враховувати, який саме підхід нейронних мереж застосовується в даний момент. Для одних моделей використання складніших архітектур чи додаткових механізмів є справді корисним і помітно покращує якість класифікації, тоді як для інших – такі ускладнення стають зайвими й не приносять суттєвого виграшу. Це означає, що вибір методів має залежати не лише від типу даних, а й від того, наскільки конкретна модель здатна ефективно використати ці додаткові можливості.

Як вже було сказано, подібні дії були проведені зі всіма датасетами, які приймають участь у дослідженні, і різниця полягає лише в деяких моментах, що залежать від структури набору та значень ознак.

Після попереднього аналізу даних кожний датасет було розділено на три частини: тренувальну, валідаційну та тестову. У таблиці 2.3 наведено кількість прикладів для кожного з цих піднаборів окремо для кожного

використовуваного датасету. Треба враховувати, що це таблиця для вже оброблених датасетів, в яких могла бути виділена певна кількість даних.

Таблиця 2.3 – Кількість записів в піднаборах всіх датасетів

Датасет	LIAR		PolitiFact		Real and Fake news		FEVER		
	0	1	0	1	0	1	0	1	2
К-сть в тренувальному набору	6600	3638	11095	4187	11136	13560	47096	114801	66380
К-сть в тестовому наборі	850	417	1958	739	2784	3390	4889	4694	6456
К-сть в валідаційному наборі	864	420	2304	869	3480	4238	4887	4638	6410

Далі, хоча це повинно бути видно ще під час EDA, ми повинні врахувати дисбаланс класів, оскільки цей фактор чинить серйозний вплив на ефективність процесу навчання. Ігнорування нерівномірного розподілу даних може призвести до зміщення моделі в бік домінуючого класу. З метою видалення цього впливу було розраховано ваги для кожного з класів за формулою 2.1.

$$w_j = \frac{N}{K * n_j}, \tag{2.1}$$

де w_j – вага для класу j;

N – загальна кількість зразків;

K – кількість класів;

n_j – кількість зразків у класі j.

Якщо розглянути всі датасети, то можна побачити, що вони всі, за виключенням Real and Fake News, мають дизбаланс класів. Тому для кожного з їх класів на основі кількості записів кожного з них в

тренувальному піднаборі було розраховано ваги. Вони представлені в таблиці 2.4.

Таблиця 2.4 – Ваги кожного класу для тренувальних піднаборів кожного датасету

Датасет	LIAR		PolitiFact		Real and Fake news		FEVER		
	0	1	0	1	0	1	0	1	2
Ваги класу	0.77	1.4	0.7	1.8	-	-	1.628	0.804	1.36

Після закінчення обробки датасетів постає питання вибору найбільш підходящого методу для розв’язання задачі. Для цього доцільно детально розглянути різні підходи, які використовуються у нейронних мережах, починаючи від тих, що зараз вважаються вже класичними і закінчуючи сучасними.

2.3 Огляд рекурентних нейронних мереж

Рекурентні нейронні мережі – доволі старий тип архітектури, який присутній в глибинному навчанні. Незважаючи на те, що зараз існують надійніші та ефективніші підходи, було свідомо вирішено включити їх у дослідження аби подивитися їх результати на задачах виявлення неправдоподібних та маніпулятивних текстів.

Рекурентні нейронні мережі – це глибокі нейронні мережі, навчені на послідовних або часових рядах даних для створення моделі машинного навчання, яка може робити послідовні прогнози або висновки на основі послідовних вхідних даних [6]. Це може бути корисно для виявлення стилістичних відхилень або маніпуляцій в текстах. Рекурентні нейронні мережі працюють за принципом збереження інформації про попередні кроки послідовності у внутрішньому стані, який оновлюється на кожному елементі вхідного ряду. Таким чином, мережа може враховувати контекст

попередніх слів або символів при обробці поточного елемента. Головна проблема полягає в тому, що базові рекурентні нейронні мережі страждають від проблеми зникаючих градієнтів, що ускладнює для них вивчення довгострокових залежностей [6]. Тому з часом було винайдено нові типи рекурентних архітектур, які намагалися вирішити цю проблему, а саме LSTM та GRU. Єдина різниця між ними – в архітектурі. LSTM використовує три типи воріт:

– forget – визначає, яку інформацію зі стану пам’яті слід забути;

– input – вирішує, яку нову інформацію додати до стану пам’яті;

– output – контролює, яку частину стану пам’яті передати на вихід мережі.

GRU, у свою чергу, має спрощену структуру з двома воротами:

– reset – вирішує, яку частину попереднього стану ігнорувати при обчисленні нового стану;

– update – контролює, скільки інформації з попереднього стану залишити та скільки додати з нового входу.

Обидва типи здатні зберігати залежності у довгих послідовностях, але GRU зазвичай демонструє схожу точність за меншої складності. LSTM часто застосовують у задачах, де критичними є складні довгострокові залежності. GRU може бути кращим вибором у випадках обмежених обчислювальних ресурсів. Виходячи з опису того як працюють та які проблеми вирішують ці два типи рекурентних мереж, можна сказати, що саме вони будуть застосовані при дослідженні.

Також, рекурентні нейронні мережі потребують значно менше обчислювальних ресурсів порівняно з сучасними моделями, що дозволяє швидко отримати перші результати. Простота їхньої архітектури робить їх зручним інструментом для побудови базового рішення, яке слугує орієнтиром для подальших експериментів. Іноді моделі, які побудовані на старих рішеннях можуть мати високу ефективність, що робить застосування сучасних підходів зайвим.

Потрібно враховувати, що рекурентні мережі запам'ятовують не факти, а правила та стилістичні патерни тексту, вивчені на тренувальних даних. Через це, якщо неправдиве твердження подано нейтрально, модель не зможе його перевірити, адже не має знань про реальний світ, а бачить лише граматично коректний текст. Однак, оскільки на практиці переважна більшість фейків для переконливості поширюється саме через маніпулятивні техніки, яку рекурентні нейронні мережі розпізнають добре, можна зробити висновок, що вони цілком підходять для визначених задач. Отже, далі потрібно зрозуміти, яка саме архітектура мережі буде найбільш ефективною. Це значною мірою буде залежити від структури та характеру датасету. Якщо в наборі даних окрім тексту присутня додаткова інформація, наприклад, про спікера, теми або заголовок новини, ці дані можуть бути включені до моделі як додаткові вхідні ознаки. У такому випадку архітектура стає багатовхідною. При ній один потік обробляє текстові послідовності через рекурентні шари, а інші потоки – додаткові ознаки, після чого вони всі об’єднуються для подальшого визначення результату. Таке поєднання дозволяє моделі одночасно враховувати контекст тексту та зовнішні фактори, що може підвищити точність передбачень. У випадку, коли датасет містить лише текст і мітку, архітектура буде максимально простою. Достатньо одного або декількох рекурентних шарів для обробки послідовності і вихідного шару для класифікації. Проста архітектура зменшує обчислювальні витрати і дозволяє дуже швидко тренувати модель, але при цьому вона має більшу ймовірність показати доволі низькі результати, хоча це також залежить від розміру датасета та даних в ньому. Також необхідно враховувати гіперпараметри, оскільки їхнє налаштування впливає на якість навчання моделі. Якщо їх значення надто великі, модель, швидше за все, перенавчиться, тобто добре запам’ятає тренувальні дані, але погано працюватиме на нових прикладах. У той же час занадто малі значення можуть призвести до недонавчання, коли модель не встигає опанувати важливі закономірності в даних.

2.4 Вплив механізму уваги на результат рекурентних нейронних мереж

Однак навіть покращені типи рекурентних нейронних мереж не завжди здатні повністю вирішити проблему довгих залежностей, особливо коли тексти містять складні структури та численні ключові фрагменти інформації. Щоб якось подолати ці обмеження, був запропонований механізм уваги.

Механізм уваги – це метод, що використовується в моделях глибокого навчання, який дозволяє моделі вибірково фокусуватися на певних областях вхідних даних під час прогнозування, не втрачаючи контексту, що особливо могло б допомогти для класифікації текстів і виявлення маніпуляцій, де критично важливі як окремі слова, так і їх взаємозв’язки [7].

Перед тим як продовжувати, потрібно розібратися як саме працює механізм уваги та які підходи він має. Спершу модель отримує вхідну послідовність у вигляді векторів. Далі обчислюються оцінки релевантності кожного елемента, які перетворюються через softmax у ваги уваги, що відображають відносну важливість елементів [8]. Потім ці ваги використовуються для обчислення зваженої суми вхідних елементів, що формує контекстний вектор – представлення найбільш важливої інформації для моделі [8]. Одним із основних підходом механізму уваги є Scaled Dot-Product. У ньому для кожного елемента обчислюються три вектори: запит, ключ та значення [8]. Оцінка релевантності елементів послідовності проводиться через скалярний добуток між запитом та ключами, результат якого ділиться на квадратний корінь розмірності ключа для покращення навчання [8]. Подальшим розвитком є Multi-Head, який обчислює увагу одночасно в кількох напрямках. Кожен напрямок аналізує різні аспекти даних, що допомагає моделі виявляти різноманітні залежності та контексти одночасно [8].

Оцінивши переваги механізму уваги, можна зробити висновок, що він потенційно дозволить моделі фокусуватися на певних словах, які

притаманні ярко вираженим фейоким текстам, а у звичайних текстах дозволить знаходити внутрішні суперечності, але в той же час, модель все не має інформації про реальний світ, тому повністю покладатися на нього в цій задачі не варто.

Для застосування механізму уваги не потрібно повністю змінювати архітектуру нейронної мережі. Зазвичай його інтегрують як проміжний шар, який обробляє виходи існуючої рекурентної моделі.

2.5 Трансформери та fine-tuning

Інтеграція механізму уваги в архітектуру рекурентних нейронних мереж може покращити роботу з контекстом у довгих реченнях, але на даний момент для багатьох завдань, які стосуються роботи з текстом, широко застосовують більш сучасні альтернативи – трансформери. Вони представляють з себе клас моделей обробки послідовностей, які замінюють поетапну рекурентну обробку механізмом самоуваги [9]. Це означає, що замість проходження тексту слово за словом кожна позиція оцінює зв’язки з усіма іншими позиціями, що дає моделі можливість напряму встановлювати глобальні залежності між віддаленими частинами тексту [10]. Для одночасного врахування різних типів зв’язків застосовується раніше згаданий multi-head attention, а на вхід подаються не лише векторні представлення токенів, а й позиційні коди, які дають інформацію про порядок слів [11]. Така архітектура дозволяє ефективно паралелізувати обчислення під час навчання, прискорюючи роботу з великими обсягами даних, а також набагато захоплювати довготривалі та складні контекстні взаємозв’язки, ніж рекурентні нейронні мережі, що робить трансформери кращим вибором для задач аналізу правдивості текстів та розпізнавання маніпулятивних патернів.

Однак, треба розуміти, що навіть найкраща архітектура мало допоможе, якщо немає достатньо даних. В деяких представлених датасетах

об’єм даних доволі невеликий, тому, щоб скористатися перевагами трансформерів при обмеженому датасеті, зазвичай застосовують fine-tuning. Він представляє з себе процес адаптації попередньо навченої моделі під конкретну задачу шляхом додаткового навчання на розмічених даних цієї задачі. Замість тренування моделі з нуля ми використовуємо вже набутий модельний «загальний» мовний досвід і тільки підлаштовуємо її під специфіку задачі, що значно знижує потребу у великій кількості даних. Є декілька способів налаштування моделей, представлено два основних:

– Full fine-tuning. Оновлюються всі параметри моделі під час навчання на цільовому датасеті [12]. За наявності достатньої кількості даних дає найвищу точність і максимальну адаптацію до завдання, але при цьому потребує багато обчислювальних ресурсів та є великий ризик перенавчання при малих наборах даних [12];

– Parameter-efficient tuning. Оновлюються лише деякі параметри або додаються нові модулі, а не вся модель [12]. При цьому підході зменшується число параметрів, що піддаються навчанню, і є можливість одночасного виконання декількох задач без дублювання великої моделі. Звісно, що при такому підході результати можуть трохи поступаються повному тонкому налаштуванню.

2.6 Використання великих мовних моделй з RAG

Fine-tuning є ефективним способом адаптації трансформерів під задачу виявлення дезінформації, але сучасні великі мовні моделі здатні генерувати зрозумілі для людини пояснення, вказуючи де саме в тексті може бути маніпуляція або підстави вважати матеріал фейком. Проте навіть найпотужніші LLM мають обмеження: вони інколи «галюціонують» – тобто формулюють упевнену, але невірну інформацію – і, що критично для фактчекінгу, їхні знання фіксовані в момент останнього навчання, тобто модель може не містити відомостей про нещодавні події або локальні

джерела [13]. Тому для надійної перевірки тверджень ефективніше поєднувати LLM із зовнішньою базою доказів. Такий підхід має назву Retrieval-Augmented Generation.

RAG – це архітектура, де генеративна модель отримує підмножину релевантних фрагментів із зовнішньої бази знань, яка використовується як контекст при формуванні відповіді [14]. На практиці у ролі такого сховища зазвичай застосовують векторну базу даних. Кожен документ або фрагмент тексту переводиться в числовий вектор, що відображає його семантичний зміст. Коли надходить новий запит або твердження, воно теж кодується в вектор, і система виконує семантичний пошук [14], потім знаходить найближчі вектори в індексі, які відповідають схожим за контекстом джерелам. Далі велика мовна модель отримує ці фрагменти як додатковий контекст і генерує зрозумілу відповідь.

Для задачі виявлення фейків цей підхід має кілька ключових переваг. По-перше, він знижує ризик галюцинацій, бо відповідь базується на конкретних доказах із бази, а не тільки на внутрішніх уявленнях моделі. По-друге, RAG робить висновки інтерпретованими. Можна побачити на які джерела опирався алгоритм або які фрази в тексті стали підставою для висновку моделі. По-третє, векторні БД дозволяють оперативно оновлювати базу, додавати свіжі данні, завдяки чому система отримує доступ до актуальної інформації без повного донавчання моделі. Що стосується стилістично виражений фейків, то тут RAG не завжди критично потрібен. Він виступає, скоріше, як корисне доповнення, оскільки такі тексті мають стилістичний, дискурсивний або прагматичний характер. Це реалізується через тон, упередженість, підсилення емоцій, підміну певних зв’язків. Сучасні великі мовні моделі ці сигнали можуть виявляти всередині самого тексту без звернення до зовнішніх джерел. Проте RAG може мати значення, коли ці тексти опираються на спотворення фактів або вибіркову подачу даних, тоді потрібні зовнішні джерела, щоби показати, які факти опущені або як їх вирвали з контексту.

3 ЗАСТОСУВАННЯ ПІДХОДІВ ТА АНАЛІЗ ЇХ РЕЗУЛЬТАТІВ

3.1 Аналіз підходів на основі рекурентних нейронних мереж

3.1.1 Підготовка до експериментів

Для подальшої роботи з текстом його потрібно подати у вигляді векторів. Однак безпосередня обробка сирого тексту створює проблеми для рекурентних нейронних мереж. Через велику кількість граматичних форм словник стає надто великим, і модель швидко перенавчається: вона запам’ятовує окремі форми слів замість загальних правил. Щоб уникнути цього та покращити здатність моделі узагальнювати, була використана лематизація – перетворення слів до їхньої базової словникової форми. Це уніфікує текст і помітно зменшує розмір словника. Такий підхід прибирає зайву граматичну варіативність й змушує модель зосереджуватися на змісті слів.

Далі було проведено перетворення тексту в цифрові послідовності, тобто токенізацію, яку вирішили виконувати на рівні слів. Цей вибір базується на особливостях рекурентних мереж. По-перше, саме слова, а не окремі символи, несуть основний зміст, потрібний для оцінки правдивості. По-друге, токенізація за словами дозволяє створювати набагато коротші вхідні послідовності. Для рекурентних мереж це критично, бо робота з коротшими ланцюжками зменшує проблему зникаючого градієнта і дозволяє моделі краще зберігати контекст тексту.

3.1.2 Результати застосування PHM на датасетах LIAR та PolitiFact

Експерименти почалися з датасету LIAR. Як уже згадувалося, він містить короткі новини або висловлювання, де оцінка правдивості часто залежить не лише від змісту, а й від того, хто саме це сказав. Тому на вхід моделі

вирішили подавати і текст, і інформацію про спікера. Архітектура нейромережі представлена на рисунку 3.1.

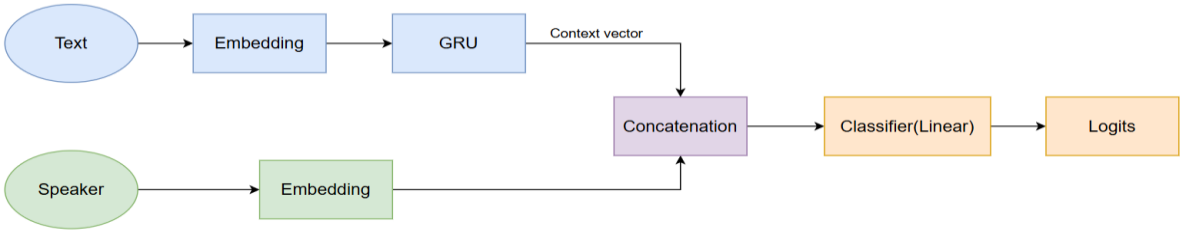

Рисунок 3.1 – Архітектура нейронної мережі для датасету LIAR

Результати метрик для кожного класу наведено в таблиці 3.1. Видно, що ця модель з самого початку зіткнулася з помітними труднощами. Ефективність вийшла посередньою. Особливо проблемною є критично низька повнота для класу правдивих новин. Нейронна мережа фактично не знаходить більшість таких записів. Це демонструє, що додавання даних про спікера не допомогло моделі якісно розділити вибірку. Очевидно, що ця архітектура не здатна виявляти складні приховані залежності у коротких текстах.

Таблиця 3.1 – Результати рекурентної моделі на датасеті LIAR

Клас	0(Fake)	1(True)
Precision	0.67	0.32
Recall	0.77	0.22
F1-Score	0.72	0.26
К-сть правильних визначень	656	90
К-сть хибних визначень	194	327
Точність		0.59

Оскільки базова GRU не продемонструвала високої точності й погано виокремлювала справді важливі ознаки тексту, наступним етапом стало додавання механізму уваги. Технічно цей підхід реалізували через введення повнозв'язного лінійного шару. Він аналізує всі приховані стани, які GRU генерує на кожному кроці, тобто для кожного токена, та обчислює їхню важливість. Потім ці значення перетворюються на ймовірності, що називаються вагами уваги. Детально архітектура представлена на рисунку 3.2. Результати представлені в таблиці 3.2

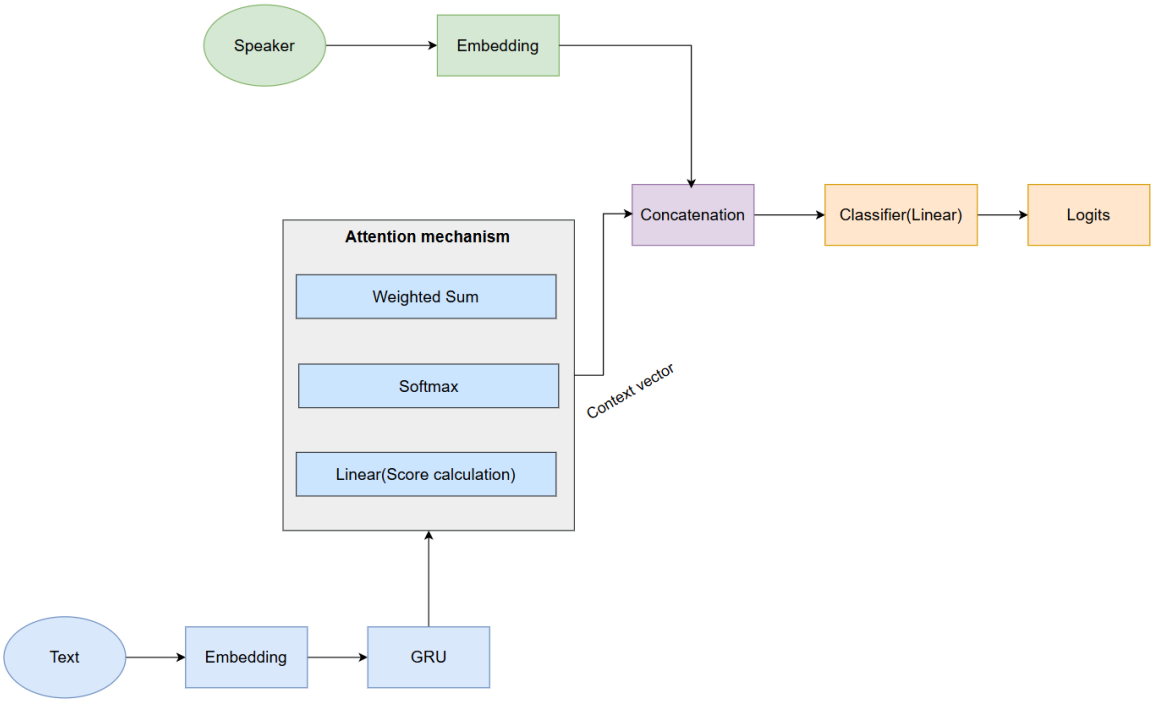

Рисунок 3.2 – Архітектура нейронної мережі з механізмом уваги для датасету LIAR

Таблиця 3.2 – Результати рекурентної моделі з механізмом уваги на датасеті LIAR

Клас	0(Fake)	1(True)
Precision	0.67	0.32
Recall	0.58	0.41
F1-Score	0.62	0.36

Продовження таблиці 3.2

Клас	0(Fake)	1(True)
К-сть правильних визначень	493	170
К-сть хибних визначень	357	247
Точність	0.52	

Загальна точність зменшилася, хоча це не критично, оскільки тут присутній дисбаланс класів, проте змінилася сама поведінка нейромережі. Найважливішим стало те, що модель значно краще почала розпізнавати правдиві новини. Якщо раніше вона майже не реагувала на цей клас, то після додавання уваги здатність виявляти його зросла майже вдвічі. Однак, модель стала частіше сприймати фейкові новини як правдиві. Через це різко зросла кількість помилкових визначень.

Далі були проведені експерименти з датасетцом PolitiFact, який, як зазначалося, є більшою і якіснішою версією набору LIAR. Була використана схожа модель, але була додана нова ознака – тип ресурсу або канал поширення інформації, наприклад блог чи телебачення. Архітектура стала трипотоковою. В ній текст оброблявся через рекурентний шар, а спікер і ресурс – через окремі ембедінги. Потім усі три вектори об’єднувалися, щоб модель робила висновок, враховуючи зміст висловлювання, особу автора та місце публікації. Схема архітектури представлена на рисунку 3.3, а результати представлені в таблиці 3.3. Можна побачити, що модель стала більш стабільною, а її здатність виявляти правдиві новини помітно зросла. Хоча вона все ще допускала багато хибних прогнозів, мережа почала краще розуміти структуру вибірки, що підтверджує ефективність гібридного підходу. Важливим чинником покращення стала й більша кількість даних у датасеті, завдяки чому модель отримала більше прикладів для навчання та змогла точніше відокремлювати правдиві й неправдиві повідомлення.

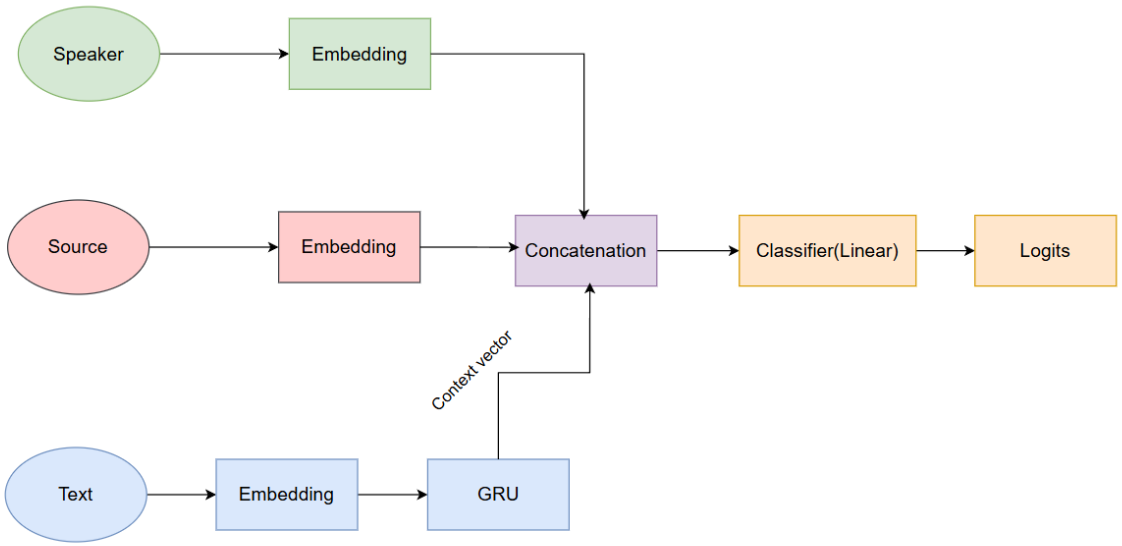

Рисунок 3.3 – Архітектура нейронної мережі для датасету PolitiFact

Таблиця 3.3 – Результати рекурентної моделі з на датасеті PolitiFact

Клас	0(Fake)	1(True)
Precision	0.83	0.40
Recall	0.62	0.68
F1-Score	0.71	0.50
К-сть правильних визначень	1212	499
К-сть хибних визначень	746	240
Точність		0.63

Далі додамо механізм уваги до цієї моделі. Схема з ним представлена на рисунку 3.4, а результат у таблиці 3.4. Видно певне покращення стабільності моделі. Порівняно з версією без механізму уваги, класифікатор став працювати обережніше й точніше, зменшуючи кількість неправильних рішень та краще розрізняючи справжні й фейкові твердження.

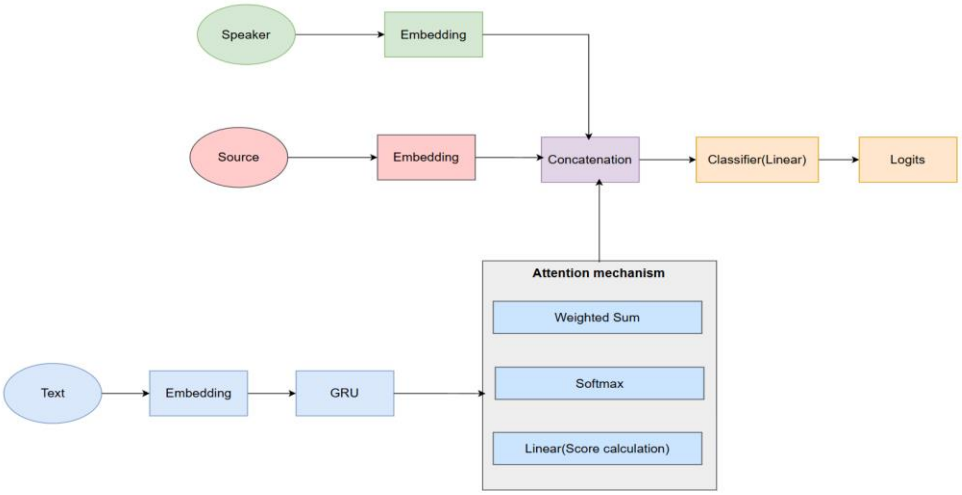

Рисунок 3.4 – Архітектура нейронної мережі з механізмом уваги для датасету PolitiFact

Таблиця 3.4 – Результати рекурентної моделі з механізмом уваги на датасеті PolitiFact

Клас	0(Fake)	1(True)
Precision	0.83	0.47
Recall	0.75	0.59
F1-Score	0.79	0.52
К-сть правильних визначень	1476	433
К-сть хибних визначень	482	306
Точність		0.70

3.1.3 Результати застосування РНМ на датасеті Real and Fake news

Наступним етапом дослідження стало тестування рекурентних мереж на датасеті Real and Fake News. Як вже говорилося, у цьому наборі правдиві новини написані офіційним, виваженим стилем від авторитетних джерел, а фейкові мають емоційний, розмовний синтаксис. Враховуючи цю

особливість, архітектуру моделі спростили до ембединга з одним рекурентним шаром, а у якості тренувальної ознаки використовується лише текст. Її схема представлена на рисунку 3.5. Цей експеримент має показати, наскільки здатні рекурентні мережі класифікувати дані, спираючись лише на лінгвістичні ознаки. Результати представлена в таблиці 3.5

Рисунок 3.5 – Архітектура нейронної мережі для датасету Real and Fake News

Таблиця 3.5 – Результати рекурентної моделі з на датасеті Real and Fake News

Клас	0(Fake)	1(True)
Precision	0.97	0.98
Recall	0.98	0.97
F1-Score	0.97	0.98
К-сть правильних визначень	2727	3303
К-сть хибних визначень	57	87
Точність		0.97

Результати тестування є дуже високими, підтвердивши важливість стилістичних ознак. Рекурентна нейронна мережа, яка аналізувала лише текст, досягла майже ідеальної точності, чітко розрізняючи стиль правдивих новин і розмовний стиль фейків. Фактично, мережа класифікувала жанри, а не перевіряла факти. Хоч результати й вражають, вони показують обмеженість датасету, бо у реальному світі дезінформація часто

приховується під нейтральним текстом. Тому використання складніших методів, як трансформери, на цьому наборі сенсу не має.

3.1.4 Результати застосування PHM на датасеті FEVER

Наступним кроком стало дослідження масштабного датасету FEVER, який був розроблений для автоматизованої перевірки фактів. Оскільки для коректної класифікації зазвичай потрібні зовнішні знання, ми спочатку вирішили провести базовий тест лише на тексті тверджень. Мета була проста: перевірити, чи може рекурентна модель вгадати вердикт, спираючись лише на те, як сформульовано речення, перш ніж будуть додано докази. У цьому випадку повторно використовується схема, що представлена на рисунку 3.5. Самі результати представлена в таблиці 3.6.

Таблиця 3.6 – Результати рекурентної моделі з на датасеті FEVER

Клас	0(Refutes)	1(Supports)	2(Not enough info)
Precision	0.67	0.43	0.52
Recall	0.50	0.43	0.62
F1-Score	0.57	0.43	0.57
К-сть правильних визначень	2425	2001	3994
К-сть хибних визначень	2464	2693	2462
Точність			0.52

Тестування базової рекурентної моделі лише на тексті тверджень показало помірну точність трохи більше 52%, доводячи, що повноцінна перевірка фактів без зовнішніх даних неможлива. Мережа не змогла відрізнити зміст і здебільшого обирала клас «Not enough info», для якого

повнота була найвищою. При цьому підтверджені та спростовані факти часто плуталися.

Далі необхідно використати доказову базу, яка представлена в датасеті у вигляді ознаки «Evidence». Перед подачею в модель її очистили, а саме: прибрали службові номери рядків та об’єднали назву статті з текстом для збереження контексту.

Архітектуру моделі змінили на паралельну мережу з спільним рекурентним шаром. В ній твердження і докази оброблялися однаково, а потім їхні вектори об’єднувалися для класифікації. Ця схема представлена на рисунку 3.6. Результати моделі представлені в таблиці 3.7.

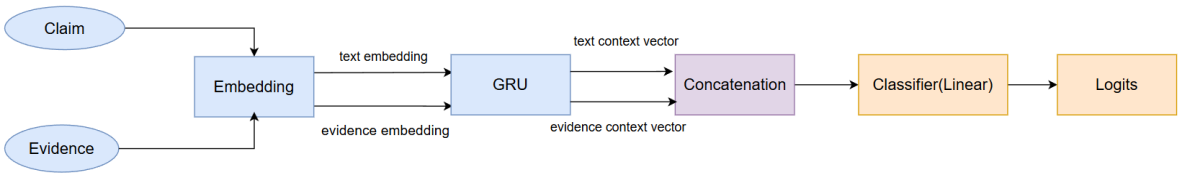

Рисунок 3.6 – Архітектура нейронної мережі для датасету FEVER

Таблиця 3.7 – Результати рекурентної моделі на датасеті FEVER

Клас	0(Refutes)	1(Supports)	2(Not enough info)
Precision	0.76	0.51	0.56
Recall	0.46	0.53	0.70
F1-Score	0.58	0.52	0.62
К-сть правильних визначень	2262	2492	4519
К-сть хибних визначень	2627	2202	1937
Точність			0.58

Результати показали невелике, але помітне зростання загальної точності, що підтвердило важливість контексту. Проте модель все ще була

сильно зміщена в бік класу «Not enough info». Це свідчить, що рекурентна мережа може оцінювати тематичну близькість, але їй бракує механізму глибокої уваги для виявлення тонких логічних зв’язків і протиріч, необхідних для повноцінної перевірки фактів.

3.1.5 Висновки на основі застосування рекурентних нейронних мереж

Рекурентні нейронні мережі, на прикладі GRU, добре показали себе у звичайних задачах класифікації текстів, а саме новин, де правдиві і фейкові матеріали помітно відрізняються за стилем написання. Вони легко вловлюють їх характерні ознаки. Однак, як вже говорилося раніше, дезінформація в реальному світі має більш комплексний вид, тому краще орієнтуватися на результати, де модель повинна спиратися саме на факти. У цьому випадку її ефективність на доволі низькому рівні. Якщо текст виглядає звичайно й не має очевидних мовних підказок, рекурентні моделі часто не здатні визначити, чи твердження є правдивим. Також було визначено, якщо моделі представити певні докази до тексту, то її результати роботи будуть трохи вище.

3.2 Аналіз підходів на основі трансформерів

3.2.1 Підготовка до експериментів

Основним інструментом для цього етапу обрали бібліотеку transformers від Hugging Face. Експерименти проводилися на трьох архітектурах: BERT, RoBERTa та DeBERTa. Кожна з цих моделей має свій словник і алгоритм токенізації. Тому дуже важливо використовувати їхні власні токенізатори. Застосування чужого призведе до великої кількості невідомих або неправильних токенів і порушить роботу моделі.

Попередньо натреновані моделі доступні в різних конфігураціях розміру – від легких версій до великих. Варіанти з більшими параметрами краще уловлюють тонкі семантичні нюанси, але вимагають значної відеопам’яті та часу на навчання, легші варіанти працюють швидше, але можуть втрачати в продуктивності на складних завданнях. У цьому дослідженні використовувалися конфігурації базового рівня, які маються баланс між продуктивністю моделі апаратними вимогами до апаратного забезпечення.

Щодо стратегії навчання, було розглянуто як повний файн-тюнінг, так і тонке налаштування лише верхніх шарів. Далі будуть наведені результати першого типу, оскільки саме цей підхід виявився ефективнішим для всіх використаних наборів даних і дозволив досягти найкращих показників.
Для деяких датасетів, як і під час експериментів з рекурентними нейронними мережами, було враховано незбалансованість класів. Тому для кожного класу налаштовано визначені ваги, щоб підсилити вплив менш представленого класу.

3.2.2 Результати застосування попередньо навченних трансформерів на датасетах LIAR та PolitiFact

Порядок датасетів залишився тим же, тому першим представлено аналіз результатів на датасеті LIAR. Було вирішено для тренування використовувати тільки текст новини, тобто без метаданих спікера. Метою було перевірити, чи може модель самостійно витягувати потрібний контекст. Результати показані в таблиці 3.8. Можна помітити прогрес в порівнянні з рекурентними нейронними мережами, бо загальна точність досягла 62%. Ще важливішим стало те, що BERT навчився краще розпізнавати правдиві новини, майже зрівнявшись із показником для фейків. Також у порівнянні з рекурентними нейронними мережами точність для правдивих новин помітно зросла, що свідчить про краще розпізнавання патернів правди.

Водночас модель все ще схильна плутати фейки з правдою, що обмежує її загальну надійність.

Таблиця 3.8 – Результати fine-tuning BERT на датасеті LIAR

Клас	0(Fake)	1(True)
Precision	0.74	0.47
Recall	0.63	0.61
F1-Score	0.68	0.53
К-сть правильних визначень	515	274
К-сть хибних визначень	303	175
Точність		0.62

Далі на цьому наборі була протестована RoBERTa. Очікувалося, що вона буде краще працювати з датасетом. Проте загальна точність залишилася на рівні BERT. Її результати представлені в таблиці 3.9. Модель стала частіше класифікувати новини як неправдиві і втрачати чутливість до достовірних. У підсумку, незважаючи на більш просунуту архітектуру, баланс між класами погіршився порівняно з базовим BERT.

Таблиця 3.9 – Результати fine-tuning RoBERTa на датасеті LIAR

Клас	0(Fake)	1(True)
Precision	0.72	0.47
Recall	0.69	0.50
F1-Score	0.70	0.48
К-сть правильних визначень	564	224
К-сть хибних визначень	254	225
Точність		0.62

Завершенням експериментів на LIAR стала модель DeBERTa. Вона продемонструвала якісний стрибок і встановила найкращий результат серед усіх перевірених моделей. Її результати показані в таблиці 3.10. Модель змогла підняти загальну точність до 65%. Однак все таки вона, як і минулі моделі, має низькі показники у визначенні правдивих новин.

Таблиця 3.10 – Результати fine-tuning DeBERTa на датасеті LIAR

Клас	0(Fake)	1(True)
Precision	0.74	0.50
Recall	0.70	0.56
F1-Score	0.72	0.53
К-сть правильних визначень	573	251
К-сть хибних визначень	245	198
Точність		0.65

Наступним етапом стало проведення експериментів на датасеті PolitiFact. Для цього набору змінилася схема подачі вхідних даних. Було вирішено використовувати метадані спікера та ресурсу. Однак, тут потрібно враховувати те, що йде робота з іншим типом нейронних мереж. Якщо в рекурентних моделях текст, спікер і джерело оброблялися окремими гілками, то трансформер потребує однієї суцільної послідовності токенів. Тому метадані перетворювалися на текст і об’єднувалися з новиною через токени [SEP]. Формат входу прийняв вигляд: [CLS] Speaker Name [SEP] Source Name [SEP] News Text.

Моделі було донавчено та протестовано в тому ж порядку, що і минулий датасет, тому розпочнемо з BERT. Результати представлені в таблиці 3.11. Загальна точність моделі зросла, причому за всіма метриками, якщо порівнювати результати з рекурентними нейроними мережами. Але

проблеми залишаються ті ж самі: модель занадто часто приймає фейкові новини за правду.

Таблиця 3.11 – Результати fine-tuning BERT на датасеті PolitiFact

Клас	0(Fake)	1(True)
Precision	0.87	0.46
Recall	0.68	0.72
F1-Score	0.76	0.56
К-сть правильних визначень	1331	532
К-сть хибних визначень	627	207
Точність		0.69

Наступною в тестуванні була, як і для минулого датасету, RoBERTa. Очікувалося, що її потужніша архітектура краще використає контекст, особливо коли до текста додатково йдуть метадані. Результати моделі представлені в таблиці 3.12. Загальна точність моделі виявилася дещо нижчою, поведінка RoBERTa трохи більше змістилася в бік високої чутливості до правдивих новин. Тому модель частіше помилково приймає фейкові новини за правду.

Таблиця 3.12 – Результати fine-tuning RoBERTa на датасеті PolitiFact

Клас	0(Fake)	1(True)
Precision	0.89	0.44
Recall	0.63	0.79
Клас	0(Fake)	1(True)
F1-Score	0.73	0.57
К-сть правильних визначень	1234	584
К-сть хибних визначень	724	155
Точність		0.67

Завершальним етапом роботи з PolitiFact стало тестування DeBERTa, результати якого представлені в таблиці 3.13. Метрики майже ідентичні тим, що і для RoBERTa.

Таблиця 3.13 – Результати fine-tuning DeBERTa на датасеті PolitiFact

Клас	0(Fake)	1(True)
Precision	0.90	0.45
Recall	0.63	0.81
F1-Score	0.74	0.58
К-сть правильних визначень	1234	599
К-сть хибних визначень	724	140
Точність		0.68

Як можна побачити, на датасетах, що містять лише звичайні новини, моделям BERT, RoBERTa та DeBERTa усе ще непросто розрізняти правду й неправду, оскільки вони навчені виявляти мовні патерни та контекстні залежності, а не звертатися до зовнішніх фактів. В той же час, трансформерні моделі все ж демонструють помітне покращення порівняно з класичними рекурентними мережами завдяки кращій здатності захоплювати довготривалий контекст і складні взаємозв’язки в текстах, але без зовнішньої верифікації їхня ефективність у задачі фактчекінгу залишається обмеженою.

3.2.3 Результати застосування попередньо навченних трансформерів на датасеті FEVER

Далі трансформери було розглянуто для задачі перевірки фактів на датасеті FEVER. Для цього була використана тільки модель DeBERTa. При

цьому, як і для рекурентних нейронних мереж, спочатоку вона була розглянута лише на текстах тверджень. Результати цього експерименту показані в таблиці 3.14. Можна побачити суттєвий прогрес порівняно з рекурентними мережами. Загальна точність зросла з 57% до 66%, що демонструє здатність трансформера ефективніше працювати в цій задачі. Модель забезпечила збалансовану роботу для всіх трьох класів. Показники повноти вирівнялися, що що свідчить про стабільність моделі. Також модель набагато рідше плутає підтверджувальні та спростувальні твердження.

Таблиця 3.14 – Результати fine-tuning DeBERTa на датасеті FEVER

Клас	0(Refutes)	1(Supports)	2(Not enough info)
Precision	0.70	0.62	0.65
Recall	0.69	0.61	0.67
F1-Score	0.69	0.62	0.66
К-сть правильних визначень	3373	2863	4326
К-сть хибних визначень	1516	1831	2130
Точність			0.66

Далі було протестовано вже з доказами. Вони були об’єднані з твердженнями у формат, який був задіяний під час роботи з датасетом PolitiFact, а саме: [CLS] Claim [SEP] Evidence. Результати представлені в таблиці 3.15. Як можна побачити, це дало найкращу точність, яка сягнула 78%, що суттєво перевищує не лише показники рекурентних мереж, але й продуктивність самої DeBERTa, коли вона тренувалась тільки на твердженнях. Таким чином стало очевидно, що ключовим фактором для верифікації фактів є доступ до релевантних доказів, а не збільшення складності архітектури. З додаткового можна сказати, що DeBERTa продемонструвала найкращу збалансованість між класами, оскільки F1-міра

має значення у межах 0.76–0.80 для всіх трьох категорій. Модель однаково впевнено визначає випадки підтвердження та спростування твердження, без перекосів, характерних для інших моделей.

Таблиця 3.15 – Результати fine-tuning DeBERTa на датасеті FEVER з використанням доказів

Клас	0(Refutes)	1(Supports)	2(Not enough info)
Precision	0.76	0.75	0.83
Recall	0.81	0.77	0.77
F1-Score	0.78	0.76	0.80
К-сть правильних визначень	3960	3614	4971
К-сть хибних визначень	929	1080	1485
Точність			0.78

3.2.4 Висновки на основі застосування трансформерів

Перехід до трансформерних архітектур, як BERT, RoBERTa та DeBERTa, дав помітний приріст точності завдяки покращеному механізму уваги та попередньому навчанні на великих обсягах даних. Проте навіть після донавчання їхні можливості залишаються обмеженими. Коли немає доступу до зовнішніх джерел, модель не дає впевненні результати в задачі виявлення дезінформції.

Натомість підхід, коли на вхід подається твердження з доказ, якість верифікації суттєво покращує. В цьому випадку, завдання перестає бути просто відтворенням запам’ятаних фактів і перетворюється на перевірку узгодженості твердження з наданими джерелами. З того, що такий підхід дав краще результати під час застованная рекурентних нейронних мереж та трансформерів, виходить, що у задачах виявлення дезінформації не можна

розраховувати виключно на внутрішні параметри моделі, а ключовим чинником успіху є наявність і якість зовнішньої доказової бази, яку модель може використовувати для визначення своїх відопідей.

3.3 Аналіз підходу застосування великих мовних моделей з RAG

3.3.1 Особливості реалізації та тестування підходу на основі LLM

Останнім підходом, який буде розглянуто в цій роботі, є використання великих мовних моделей у поєднанні з підходом RAG. Поєднання цих двох технологій виглядає доволі перспективною у задачі виявлення дезінформації, оскільки сучасні мовні моделі забезпечують значно глибший і точніший аналіз контексту та семантики, ніж рекурентні нейромережі, а також ніж базові трансформерні моделі, а механізм пошуку дозволить системі надавати їй доступ до додаткових актуальних або нещодавно з’явлених даних, яких не було під час навчання моделі.

Однак, для тестування цього підходу на локальних пристроях, таких як звичайні комп’ютери чи ноутбуки, виникають певні труднощі. Варто зазначити, що повноцінне навчання моделей подібного масштабу є надзвичайно ресурсномістким завданням. Воно потребує великих обчислювальних потужностей, величезних обсягів навчальних даних та значних часових витрат, що є проблематичним для локальних досліджень. В той же час, сучасний розвиток у цій сфері пропонує широкий вибір уже навчених моделей – як комерційних, так і таких, що мають відкритий вихідний код. Кожна з них володіє власними архітектурними особливостями, які забезпечують певні переваги, але також мають і свої обмеження. У зв’язку з цим, метою даного розділу не є проведення порівняльного аналізу ефективності десятків різних моделей. Основна увага буде зосереджена на демонстрації та дослідженні можливостей самого підходу до перевірки фактів. Для тестування буде використано одну

відкриту модель – gpt-oss-20b, розгорнуту локально. Ця модель обрана через свій великий масштаб та доволі непоганими результатами, як для локальної моделі, що дозволяє адекватно оцінювати можливості системи детекції дезінформації без необхідності зовнішніх платних сервісів [15]. Незважаючи на те, що модель розгорнута локально, її використання все одно потребує значних обчислювальних ресурсів та часу. Процес генерації відповіді для великих мовних моделей значно повільніший порівняно з класичними класифікаторами, а вимоги до пам’яті та процесорного часу залишаються високими. Через це для тестування було скорочено обсяг тестової вибірки.

Слід зауважити, що у реальних умовах розробки доцільно проводити порівняння різних архітектур із використанням специфічних для них метрик, таких як оцінка релевантності контексту, зв’язності відповіді або рівня «галюцинацій». Однак, у рамках даного дослідження вихідний формат роботи моделі буде той самий, що і у попередніх підходах. Система буде налаштована на повернення виключно міток класів.

3.3.2 Реалізація підходу RAG із використанням пошуку в інтернеті

Першим сценарієм використання великих мовних моделй з RAG, розглянутим у цьому розділі, є архітектура, де роль сховища знань виконує не статична векторна база даних, а пошукова система, у даному випадку – Google Search. Необхідність такого рішення зумовлена специфікою задачі виявлення дезінформації. Як вже було сказано, великі мовні моделі мають фіксовану дату закінчення навчання, після якої вони не володіють інформацією про події у світі. Оскільки фейкові новини часто стосуються актуальних подій, ізольована модель не здатна ефективно перевірити твердження, що з'явилися після її тренування, і схильна до «галюцинацій», які можуть мати вигляд генерації правдоподібних, але неправдивих відповідей. Крім того, можливі ситуації, коли необхідно перевірити старі

новини або факти, що були актуальні раніше, а в моделі знання про це відсутні. У такому випадку також потрібен доступу до зовнішніх джерел інформації.

У запропонованому підході процес отримання відповіді від моделі відбувається наступним чином:

– отримання твердження. На вхід системи подається текст новини, заголовок або конкретне твердження, яке потрібно перевірити;

– пошук доказів. На цьому етапі система формує пошуковий запит на основі отриманого твердження. Через API звертається до обраної пошукової системи і отримує список релевантних фрагментів тексту з веб-сторінок, які можуть містити підтвердження або спростування твердження;

– підготовка промпту. Отримані докази та оригінальне твердження об’єднуються в єдиний промпт та подаються на вхід мовній моделі;

– аналіз та формування відповіді. Модель оцінює відповідність твердження знайденим фактам і генерує результат у вигляді мітки. Також, за бажанням, можна налаштувати відповідь таким чином, щоб можна було бачити роз’яснення моделі чому вона вирішила відповісти саме так.

Для технічної реалізації етапу пошуку доказів у даному експерименті було застосовано Google Fact Check Tools API. Він забезпечує доступ до глобальної бази даних вердиктів, сформованої професійними фактчекінговими організаціями та новинними агентствами [16]. Такий вибір джерела забезпечує суттєве підвищення якості контексту для великої мовної моделі. На відміну від звичайного веб-пошуку, який повертає сирі та неструктуровані результати, які до того ж можуть бути фейковими, вибраний сервіс надає вже верифіковану інформацію з високим рівнем довіри, що знижує ризик генерації хибних або неперевірених даних. Це особливо важливо для задачі виявлення дезінформації, де критично необхідно опиратися на достовірні джерела.

Звісно, незважаючи на свої сильні сторони, сервіс має один суттєвий недолік: він не охоплює всі можливі твердження або новини. Деякі події

можуть бути занадто новими, локальними або рідко висвітленими, через що Google Fact Check Tools API не завжди повертає релевантні результати. Тому в архітектуру системи додатково закладено механізм резервного пошуку. У випадках, коли спеціалізований API не знаходить підтверджень або спростувань для заданого твердження, система автоматично перемикається на виконання загального пошукового запиту у відкритому вебі. Такий підхід дозволяє підтримувати працездатність системи навіть у ситуаціях із обмеженою наявністю фактчекінгових даних, забезпечуючи отримання додаткових джерел інформації для аналізу. Як вже зазначалося вище, такий тип пошуку без обмежень джерел несе ризики, оскільки у видачу можуть потрапляти ресурси з низькою репутацією або сайти, що поширюють фейки. У реальних проектах для вирішення цієї проблеми доцільно налаштовувати фільтрацію джерел, обмежуючи пошук заздалегідь визначеними авторитетними джерелами, які можуть бути релевантними для задачі. Таке обмеження простору пошуку суттєво підвищило б якість вхідних даних. Однак у рамках даної роботи пошук буде здійснюватися по всьому інтернету.

Схематичне представлення цього робочого процесу показано на рисунку 3.7.

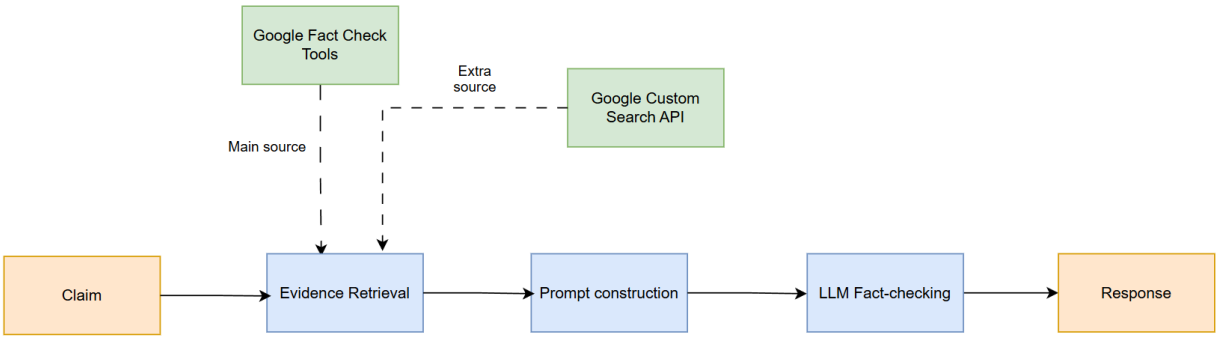

Рисунок 3.7 – Робочий процес RAG через пошук в інтернеті

Не менш важливим аспектом роботи з великими мовними моделями є інженерія промптів, яка дозволяє тонко налаштовувати поведінку системи.

За допомогою ретельно сформульованих інструкцій можна задати бажану поведінку, встановити специфічні критерії оцінювання, визначити рівень суворості при аналізі інформації та контролювати тон відповіді. Для взаємодії з мовною моделлю було використано два типи підказок. Перший – системний промпт, яка визначає роль моделі як суворого фактчекера і накладає критичні обмеження на її поведінку: робити висновки виключно на основі наданих доказів. Також зазначено, які мітки повинна повернути модель. Його текст представлений на рисунку 3.8.

```
system_prompt = (
    "You are a strict AI fact-checker. Your task is to determine whether the news item is true (True) or fake (Fake), "
    "based EXCLUSIVELY on the provided evidence."
)
```

Рисунок 3.8 – Системна підказка для моделі в першому експерименті

Другий тип – це промпт користувача, яка динамічно об’єднує вхідне твердження та знайдені джерела інформації і передає їх моделі у вигляді одного структурованого запиту. Він містить алгоритмічні вказівки для класифікації, чітку логіку прийняття рішення і вимогу виводити відповідь зазначеному форматі. Текст підказки представлений на рисунку 3.9

```
user_prompt = f"""
### INPUT DATA
CLAIM: "{claim}"

FOUND EVIDENCE (CONTEXT):
{evidence if evidence else "No external information found."}

### INSTRUCTIONS
1. Analyze the evidence.
2. If the evidence confirms the claim -> verdict 1 (True).
3. If the evidence refutes the claim -> verdict 0 (Fake).
4. If there is no evidence or it is unrelated -> verdict 0 (consider it questionable).

### RESPONSE FORMAT
Return ONLY a JSON object with no extra words:
{{
    "reasoning": "Brief explanation (1 sentence)",
    "label": 1 or 0
}}
"""
```

Рисунок 3.9 –Підказка користувача для моделі в першому експерименті

Для тестування запропонованого підходу було відібрано 70 записів кожного типу з датасета PolitiFact. Вибірка була сформована рівномірно: по 35 прикладів із категорії правдивих тверджень та по 35 – із категорії неправдивих. Результати представлено в таблиці 3.16.

Таблиця 3.16 – Результати тестування роботи LLM з RAG через пошук в інтернеті

Клас	0(Fake)	1(True)
Precision	0.63	0.81
Recall	0.89	0.49
F1-Score	0.74	0.61
К-сть правильних визначень	31	17
К-сть хибних визначень	4	18
Точність		0.69

Отримані результати демонструють, що розроблена модель намагається уникати помилкових підтверджень. За метриками можна побачити, що коли система класифікує інформацію як правдиву, ймовірність помилки є мінімальною. Водночас, як показав аналіз її обґрунтувань, модель схильна відхиляти твердження, що містять часткову правду або нюанси, які не повністю збігаються зі знайденими доказами, маркуючи їх як недостовірні задля безпеки. Така поведінка значною мірою залежить від налаштувань моделі, а саме від системного промпту. Саме закладені в ньому інструкції визначають чи буде модель інтерпретувати неоднозначні або частково підтверджені факти як істину, чи діятиме максимально консервативно. Це також підкреслює, що в подібних задачах критерії прийняття рішень можуть бути гнучко адаптовані шляхом коригування інструкцій системного промпту відповідно до вимог щодо чутливості системи.

3.3.3 Реалізація підходу RAG із використанням векторної бази даних

Наступним етапом дослідження стала реалізація архітектури RAG з використанням векторних баз данних. На відміну від попереднього підходу, де модель отримувала інформацію з відкритого інтернету, у цьому випадку вона працює в середовищі, де вся доступна інформація заздалегідь визначена. Такий підхід особливо корисний для виявлення дезінформації в закритих або специфічних сферах, де потрібно аналізувати дані, що стосуються конкретної організації, галузі чи регіону, без впливу зовнішніх неперевірених джерел.

Для цього експерименту було обрано датасет FEVER. Для формування внутрішнього сховища знань були використані текстові докази з його тренувальної вибірки. Усі фрагменти було векторизовано – перетворено на числові ембеддинги – та завантажено у локальну векторну базу даних. Оскільки до сховища увійшла вся тренувальна частина датасету, обсяг таких записів формує достатньо великий та різноманітний пошуковий простір. Це створює умови, за яких системі потрібно знаходити релевантний факт серед сотень тисяч інших записів, що добре моделює реальні сценарії роботи векторних сховищ знань.

Тестування проводиться на окремій підвибірці з 230 записів, взятих з тестової частини датасета. Докази з цієї вибірки не потрапляли до бази даних, що дозволяє коректно оцінити роботу моделі в умовах, які наближені до реальних. Щоб оцінити реальний вплив механізму RAG на якість детекції, було проведено два тестування:

– модель отримує на вхід лише текст твердження і повинна визначити його правдивість, спираючись виключно на власні знання, отримані під час навчання;

– до кожного твердження додається контекст, знайдений у векторній базі даних. Модель повинна зіставити твердження зі знайденими фактами і винести вердикт.

Для обох експериментів було налаштовано системний промпт, який визначає поведінку моделі при класифікації тверджень. У першому моделі говорилося працювати без зовнішнього контексту і визначати мітки, спираючись лише на себе. У другому експерименті системний промпт, який у якості приклада представлений на рисунку 3.10, орієнтував модель на використання інформації з локальної векторної бази знань, де контекстна інформація позначалася як абсолютна істина, а модель мала зіставляти твердження зі знайденими доказами, спираючись виключно на наданий контекст. Результати першого експерименту було представлено в таблиці 3.17.

```
system_prompt_fever = (
    "You are a strict logical verification engine, NOT a chat assistant. Your task is to perform Fact Verification.\n"
    "Your goal is to determine the relationship between the Input Claim and the provided Context Information.\n"
    "--------------------\n"
    "Context Information (This is the absolute GROUND TRUTH):\n"
    "{context_str}\n"
    "--------------------\n"
    "Input Claim to verify: {query_str}\n"
    "--------------------\n"
    "INSTRUCTIONS:\n"
    "1. Rely ONLY on the Context Information above. Treat the context as a closed knowledge source.\n"
    "2. Determine the correct label based on the standard FEVER criteria:\n"
    "   - **SUPPORTED**: If the Context explicitly states or logically implies the Input Claim.\n"
    "   - **REFUTES**: If the Context explicitly contradicts the Input Claim.\n"
    "   - **NOT ENOUGH INFO**: If the Context does not provide sufficient information to prove or disprove the Input Claim.\n"
    "3. DO NOT output any other words, explanations, or phrases.\n"
    "4. Answer with ONE of the following THREE WORDS ONLY: SUPPORTED, REFUTES, or NOT ENOUGH INFO.\n"
    "\n"
    "Answer:"
)
```

Рисунок 3.10 – Системна підказка для моделі в експерименті з векторним RAG

Таблиця 3.17 – Результати тестування роботи LLM без RAG на підвиборці тестової вибірки датасету FEVER

Клас	0(Refutes)	1(Supports)	2(Not enough info)
Precision	0.37	0.76	0.45
Recall	0.78	0.75	0.08
F1-Score	0.50	0.75	0.13
К-сть правильних визначень	39	86	5
К-сть хибних визначень	11	29	60
Точність			‘0.57

Аналіз показує, що без додаткового контексту модель майже не розпізнає ситуації, коли їй бракує інформації. Вона часто намагається «вгадати» і схиляється маркувати твердження як неправдиві, навіть якщо факти невідомі. Через це виникає багато помилкових визначень фейків. Для справді неправдивих тверджень модель поводиться надто агресивно, показуючи високу впевненість, але низьку точність. Твердження, які потребують відповіді «недостатньо інформації», практично не розпізнаються. В результаті загальна якість класифікації залишається низькою і лише трохи краща за випадкове вгадування. Це показує, що без зовнішнього контексту модель не може ефективно виконувати роль фактчекера.

Результати другого експеримента представлені в таблиці 3.18.

Таблиця 3.18 –Результати тестування роботи LLM без RAG на підвиборці тестової вибірки датасету FEVER

Клас	0(Refutes)	1(Supports)	2(Not enough info)
Precision	0.62	0.86	0.64
Recall	0.66	0.77	0.68
F1-Score	0.64	0.81	0.64
К-сть правильних визначень	33	89	44
К-сть хибних визначень	17	26	21
Точність			0.72

Впровадження механізму RAG змінило поведінку моделі на тій самій вибірці. Найсуттєвіший ефект спостерігається для випадків, коли інформації недостатньо, бо модель почала коректно ідентифікувати такі твердження замість того, щоб вгадувати або генерувати неправильні відповіді. Завдяки наданому контексту з бази знань система змогла більш впевнено розрізняти

ситуації з браком даних. Загальна точність класифікації значно покращилася, що свідчить про підвищену здатність моделі правильно верифікувати правдиві факти. Також зменшилася кількість хибних спростувань: система стала обережнішою при маркуванні тверджень як неправдивих. Для класу неправдивих тверджень точність виросла, і модель рідше плутала його з іншими категоріями. В цілому RAG дозволив моделі ефективніше використовувати зовнішні знання, підвищуючи надійність вердиктів.

3.3.4 Висновки на основі застосування великих мовних моделей із підходом RAG

Підсумок експериментів показав, що наявність контексту – це важливіша для успіху фактчекінгу компонента, ніж лише внутрішні знання моделі. Великі мовні моделі без доступу до зовнішніх фактів часто галюціюють і не вміють усвідомлювати власну необізнаність, генеруючи правдоподібні, але необґрунтовані відповіді. Після впровадження RAG модель вже не просто генерує текст на основі своїх ваг, а приймає рішення, опираючись на надані докази. На данний момент, навіть сучасні архітектури не замінять саму наявність достовірних даних. Більш просунуті та великі моделі можуть давати кращі результати і підвищувати точність аналізу, але вони теж потребують ґрунтування на надійних джерелах.

ВИСНОВКИ

Під час виконання роботи був проведений аналіз того, що представляє з себе термін «дезінформація» та які форми вона набуває. Було прийняте рішення розглядати цю проблему в контексті двох задач, а саме визначенні неправдивих текстів або тверджень на основі доказів. Далі були вибрані набори даних, які спеціалізуються якраз на них. З них 3 призначені для класифікації правдивих та неправдивих новин, а 1 – для визначення тверджень, де потрібно спиратися на додаткові докази. Також для цих наборів був проведений попередній аналіз даних, щоб було зрозуміло, які ознаки повинні приймати участь у тренуванні моделей. Крім того, було вибрано основні метрики, які будуть використовуватися для оцінки ефективності моделей.

Далі був проведений огляд та аналіз підходів нейронних мереж, як старих, так і сучасних. Серед першої категорії були розглянути рекурентні нейронні мережі, чому їх можна використати для вирішення поставлених задач та які недоліки вони мають. Далі був розглянутий механізм уваги, як потенційне покращення рекурентних мереж завдяки якому модель може краще працювати з довгими текстовими послідовностями. Після цього було перейдено до оцінки сучасних підходів, а саме трансформерів. Було визначено, щоб повністю застосувати всі переваги цієї архітектури доцільно брати заздалегідь треновану модель і донавчати її на вибраних датасетах. Тобто застосувати процес fine-tuning, який у свою чергу має два основні типи. Кінцевим об’єктом огляду, був розглянутий такий підхід як RAG, який дозволяє використовувати натреновані великі мовні моделі, додаючи інформацію з зовнішніх джерел, яким можуть виступати, наприклад, бази даних або інтернет.

Усі вищезазначені підходи були детально розглянуті та проаналізовані. Рекурентні нейронні мережі продемонстрували високі результати в тих

випадках, коли правдиві й фейкові тексти були розділені за стилем написання, але їхня ефективність суттєво знижувалася, коли потрібно було класифікувати лише за вмістом тексту та з додатковими метаданими, проте вже на етапі зіставлення фактів RNN показувала відносно кращі здібності до виявлення невідповідностей. Трансформери в цілому дали кращі результати порівняно з RNN, хоча ефективність залишається посередньою. У задачах зіставлення фактів вони працюють дуже непогано і забезпечують більш стабільну якість. Підхід RAG було досліджено в двох варіантах: з використанням інтернету як зовнішнього сховища та з застосуванням векторної бази даних. Хоча представлені експерименти виконувалися на невеликих обсягах даних і з не найбільш сучасними моделями, спостерігається помітне покращення продуктивності при додаванні зовнішньої інформації.

Загалом нейронні мережі справді можна застосовувати для виявлення дезінформації як ефективний інструмент, але для надійної роботи потрібна велика доказова база та додаткові джерела підтвердження, оскільки покладатися тільки на модель не варто, тому її результати слід поєднувати з перевіркою фактів і людською експертизою.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Disinformation and 7 Common Forms of Information Disorder. The Commons. URL: https://commonslibrary.org/disinformation-and-7-common-forms-of-information-disorder (дата звернення: 30.10.2025).

2. The seven categories of online disinformation | RSF Resources for Journalists. *RSF Resources for Journalists*. URL: https://safety.rsf.org/the-seven-categories-of-online-disinformation (дата звернення: 30.10.2025).

3. Infodemic. *World Health Organization (WHO)*. URL: https://www.who.int/health-topics/infodemic/the-covid-19-infodemic (дата звернення: 31.10.2025).

4. The Algorithmic Echo Chamber: How Curated Content Fuels Misinformation. *Uniting for Global Cyber Resilience | CyberPeace*. URL: https://www.cyberpeace.org/resources/blogs/the-algorithmic-echo-chamber-how-curated-content-fuels-misinformation (дата звернення: 31.10.2025).

5. What is Tokenization in Natural Language Processing (NLP)? - GeeksforGeeks. *GeeksforGeeks*. URL: https://www.geeksforgeeks.org/nlp/tokenization-in-natural-language-processing-nlp (дата звернення: 04.11.2025).

6. Stryker C. What is a Recurrent Neural Network (RNN)? | IBM. *IBM*. URL: https://www.ibm.com/think/topics/recurrent-neural-networks (дата звернення: 05.11.2025).

7. Tioluwani O. What are Attention Mechanisms in Deep Learning?. freeCodeCamp.org. URL: https://www.freecodecamp.org/news/what-are-attention-mechanisms-in-deep-learning (дата звернення: 06.11.2025).

8. Attention Is All You Need. *arXiv.org*. URL: https://arxiv.org/abs/1706.03762 (дата звернення: 06.11.2025).

9.Deep Learning for Natural Language Processing (DL4NLP) | Chapter 3: Transformer. URL: https://slds-lmu.github.io/dl4nlp/chapters/03_transformer (дата звернення: 08.11.2025).

10. Sayyed J. Transformers in NLP Explained—From RNNs to Attention and Beyond. Medium. URL: https://medium.com/@sjavedsayyed346/transformers-in-nlp-explained-from-rnns-to-attention-and-beyond-a571f76acfb6 (дата звернення: 10.11.2025).

11. Ferrer J. How Transformers Work: A Detailed Exploration of Transformer Architecture. DataCamp. URL: https://www.datacamp.com/tutorial/how-transformers-work (дата звернення: 10.11.2025).

12. Fine-tuning LLMs: overview and guide. Google Cloud. URL: https://cloud.google.com/use-cases/fine-tuning-ai-models (дата звернення: 14.11.2025).

13. Contributors to Wikimedia projects. Knowledge cutoff - Wikipedia. Wikipedia, the free encyclopedia. URL: https://en.wikipedia.org/wiki/Knowledge_cutoff (дата звернення: 17.11.2025).

14. What is Retrieval Augmented Generation (RAG)? | A Comprehensive RAG Guide. Elastic – The Search AI Company | Elastic. URL: https://www.elastic.co/what-is/retrieval-augmented-generation (дата звернення: 17.11.2025).

15. Introducing gpt-oss, URL: https://openai.com/index/introducing-gpt-oss (дата звернення: 29.11.2025)

16. Fact Check Tools API | Google for Developers. Google for Developers. URL: https://developers.google.com/fact-check/tools/api (дата звернення: 30.11.2025).