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 інтелектуального аналізу медіаконтенту для визначення ключових інформаційних маніпуляцій
(тема)

Виконав:
здобувач другого року навчання,
групи СШІм-23-2
Олександра Харіна
(власне ім'я, прізвище)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)
Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)
Освітня програма Системи штучного інтелекту
(повна назва освітньої програми)
Керівник доц. Марія Головянко
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ШІ _____ Олег ЗОЛОТУХІН
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Штучного інтелекту

Рівень вищої освіти другий (магістерський)

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

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

Освітня програма Системи штучного інтелекту
(повна назва)

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

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Харіні Олександрі Сергіївні
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів інтелектуального аналізу медіаконтенту для визначення ключових інформаційних маніпуляцій

затверджена наказом університету від 21 квітня 20 25 р. № 295Ст

2. Термін подання студентом роботи до екзаменаційної комісії 6 червня 20 25 р.

3. Вихідні дані до роботи Науково-технічні публікації, дані Інтернет-джерел та наукових досліджень.

4. Перелік питань, що потрібно опрацювати в роботі

1) Аналіз предметної галузі

2) Проєктування рішення

3) Експериментальне дослідження

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	21.04.2025	виконано
2	Пошук та аналіз існуючих досліджень	22.04.2025 – 31.04.2025	виконано
3	Вивчення теоретичного підґрунтя	01.05.2025 – 09.05.2025	виконано
4	Проєктування рішення	10.05.2025 – 16.05.2025	виконано
5	Проведення експерименту	17.05.2025 – 20.05.2025	виконано
6	Створення пояснювальної записки	21.05.2025 – 04.05.2025	виконано
7	Захист	10.06.2025	

Дата видачі завдання 21 квітня 20 25 р.

Здобувач О. [redacted]
(підпис)

Керівник роботи _____ доц. Марія Головянко
(підпис) (посада, власне ім’я, прізвище)

РЕФЕРАТ

Пояснювальна записка: 76 с., 18 рис., 3 табл., 1 дод., 48 джерел.

ГЛИБИННЕ НАВЧАННЯ, ГРАФ ЗНАНЬ, ЛІНГВІСТИЧНА ОБРОБКА ТЕКСТУ, МУЛЬТИАГЕНТНІ СИСТЕМИ, НЕЙРОННІ МЕРЕЖІ, РОЗПІЗНАВАННЯ ФЕЙКОВИХ НОВИН, BERT, HAT, NLP.

Об’єкт дослідження – задача розпізнавання дезінформації та маніпуляцій у медіа-контенті.

Предмет дослідження – методи інтелектуального аналізу тексту для розпізнавання різних видів дезінформації.

Мета роботи – проєктування системи для розпізнавання дезінформації.

Методи дослідження – аналіз існуючих рішень з використанням класичних алгоритмів машинного навчання та глибинних нейронних мереж для задачі розпізнавання текстових інформаційних маніпуляцій; огляд та генералізація існуючих наборів даних; моделювання архітектури рішення.

В результаті виконання кваліфікаційної роботи було проведено аналіз існуючих наукових досліджень та публікацій щодо методів розпізнавання дезінформації шляхом машинного навчання. Було виділено основні переваги та недоліки існуючих підходів та популярних наборів даних, також було запропоновано гібридний підхід для вирішення поставленої задачі, що поєднує аналіз внутрішніх даних за допомогою моделі з увагою та аналіз зовнішнього контексту на основі графу знань та гетерогенної графової мережі з увагою.

ABSTRACT

Master’s thesis contains: 76 pp., 18 fig., 3 tabl., 1 ann., 48 references.

BERT, DEEP LEARNING, FAKE NEWS DETECTION, HAT, KNOWLEDGE GRAPH, LINGVISTIC TEXT ANALYSIS, MULTIAGENT SYSTEM, NEURAL NETWORK, NLP.

The object of the study is the problem of recognizing disinformation and manipulation in media content.

The subject of the study is text mining methods for recognizing various types of disinformation.

The purpose of the work is to design a system for recognizing disinformation.

Research methods are analysis of existing solutions using classical machine learning algorithms and deep neural networks for the problem of recognizing text information manipulations; review and generalization of existing data sets; modeling of the solution architecture.

As a result of the qualification work, an analysis of existing scientific research and publications on methods for recognizing disinformation through machine learning was conducted. The main advantages and disadvantages of existing approaches and popular data sets were highlighted, and a hybrid approach was also proposed to solve the problem, combining analysis of internal data using a model with attention and analysis of the external context based on a knowledge graph and a heterogeneous graph network with attention.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ГЗ – граф знань;

МАС – мультиагентна система;

ШІ – штучний інтелект;

AI – Artificial Intelligence – штучний інтелект;

API – Application Programming Interface – інтерфейс програмування застосунків;

BOW – Bag Of Words – алгоритм «мішок слів»;

CBOW – Continuous Bag Of Words – безперервний «мішок слів»

CNN – Convolutional Neural Network – згорткова мережа;

DL – Deep Learning – глибинне навчання;

GAN – Generative Adversarial Network – генеративна змагальна мережа;

GAT – Graph Attention Network – графова нейронна мережа з механізмом самоуваги;

GCN – Graph Convolutional Networks – графова згорткова мережа;

GNN – Graph Neural Network – графова нейронна мережа;

HAT – Heterogeneous Graph Attention Network – гетерогенна графова нейронна мережа з механізмом самоуваги;

LLM – Large Language Model – велика мовна модель;

LSTM – Long-Short-Term Memory – довго-коротко-строкова пам’ять;

ML – Mashing Learning – машинне навчання;

NLP – Natural Language Processing – обробка природної мови;

RAG – Retrieval-Augmented Generation – генерація з доповненням через пошук;

RNN – Recurrent Neural Network – рекурентна нейронна мережа;

SVM – Support Vector Machine – метод опорних векторів.

ВСТУП

Ми живемо в епоху безмежного доступу до інформації, де новинні стрічки та соціальні мережі автоматично подають нам актуальні новини, поради та навчальні матеріали. Проте ця практична зручність має значний недолік: прагнучи заощадити власні зусилля, люди часто ігнорують необхідність перевірки актуальності та достовірності даних, а також надійності їхніх джерел.

Цей контекст ускладнюється тим, що створення та поширення власного контенту теж є легким і майже необмеженим: будь-хто з підключенням до інтернету може викласти та поширити будь-яку думку. Таким чином вся відповідальність за правдивість інформації лягає виключно на її автора, чия совість не є надійним гарантом.

Таким чином, ми спостерігаємо збільшення поширення неперевірених даних та зростаючу схильність людей вірити майже всьому, що вони бачать в інтернеті – ідеальні умови для поширення дезінформації (фейків), шкідливих соціальних ідей та маніпулятивного контенту. Reuters Institute у своєму глобальному опитуванні за 2024 рік [1] виявив, що 59% респондентів стурбовані здатністю відрізняти «реальне від фейкового» в інтернеті, коли йдеться про новини, цей показник зріс на 3% від значення за 2023 рік.

Існує декілька інструментів протидії дезінформації:

– медіаграмотність (навички та знання, що дозволяють ефективно і безпечно користуватися медіа сервісами);

– фактчекінг (перевірка достовірності інформації);

– контроль поширення дезінформації на законодавчому рівні або через саморегуляцію медіа [2].

Особливої уваги заслуговує фактчекінг, оскільки він є невід’ємною складовою всіх перелічених рішень. Перевірка інформації переважно потребує багато часу, а також специфічні знання та вміння, що в контексті

великої кількості даних створює запит на автоматизацію цього процесу за допомогою інструментів машинного навчання.

Останніми роками в цій галузі ведеться багато досліджень та розробок, демонструючи хороші результати. Проте існуючі рішення не є досконалими, та мають низку слабких місць, до того ж така складна задача потребує постійного оновлення з точки зору нових, більш актуальних алгоритмів та технік машинного навчання (англ. ML). Тож ця робота широко досліджує існуючі методи інтелектуального аналізу та обробки переважно текстових медіа-даних, з точки зору їх застосування для виявлення дезінформації.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

Дезінформація – це спотворена, свідомо неправдива, провокаційно-тенденційна інформація, поширена як правдива з метою введення в оману громадськості, політичних опонентів, конкурентів тощо[3]. У звіті всесвітнього економічного форуму (ВЕФ) у Давосі 2024 року [4] дезінформація знаходиться на першому та п’ятому місцях у рейтингу загроз світовій економіці протягом двох і десятьох років відповідно (рисунок 1.1).

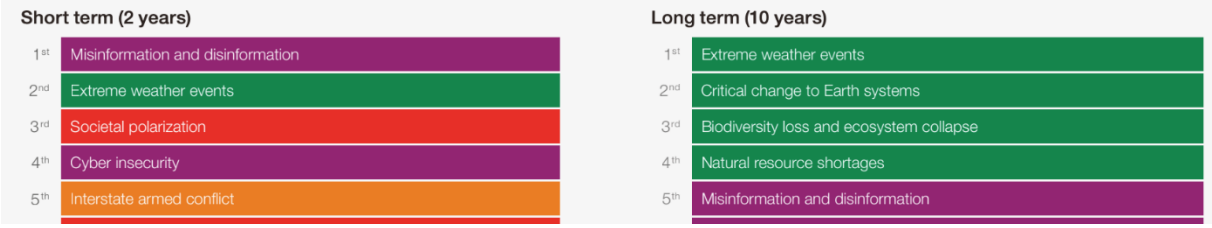

Рисунок 1.1 – Візуалізація рейтингу загроз світові економіці згідно зі звіту ВЕФ 2024 року [4]

Задача розпізнавання дезінформації потребує її чіткої класифікації, оскільки категоріальні особливості та відмінності можуть впливати на вибір інструментів та конструювання навчальних даних. Загалом можна визначити наступні 5 категорій дезінформації:

– сфабрикований контент: повністю неправдиві новини, створені з метою обману та завдання шкоди;

– маніпулятивний контент: справжня інформація або зображення, якими маніпулюють, щоб ввести читача в оману;

– неправдивий контекст: поширення справжнього контенту з неправдивою контекстною інформацією;

– хибний зв’язок: заголовки, візуальні ефекти та підписи, які суперечать або не підтверджують зміст новини;

– сатира або пародія: не має наміру завдати шкоди, але може обдурити читача, який не перевіряє посилання [5].

Загалом дослідження в цій галузі надають перевагу бінарній класифікації, маючи на увазі, що інформація або правдива, або ні. В деяких окремих випадках використовується дискретна градація брехні від незначної неправди до абсолютної дезінформації [6]. Окремі роботи концентруються на визначенні сатири в тексті поза контекстом розпізнавання брехні. Враховуючи те, що багато комедійних та фентезійних текстів містять в собі брехню можна стверджувати, що рішення проблеми розпізнавання дезінформації має відрізняти маніпулятивну брехню від жартів, і недостатня увага о цього аспекту є одним зі слабких місць існуючих досліджень.

Ще однією важливою особливістю даних для цієї задачі є залежність якості розпізнавання від різноманітності галузей інформації представлених в навчальних даних, оскільки елементи, що використовуються для визначення брехні в журналістській статті, технічному тексті та в пості з соціальних мереж переважно відрізняються [7]. Більшість існуючих розробок з питання визначення дезінформації намагається враховувати цей аспект поєднуючи набори даних різних стилістик та тематик.

Задача розпізнавання неправдивої інформації в медіапросторі є дуже обширною та має різноманітні формати вхідних даних для аналізу. Загалом можна виокремити наступні джерела інформації:

– наповнення контенту: текст інформації, заголовок, зображення, автор та інформація про нього, дата публікації, коментарі та інші метадані;

– контекст: соціально-культурні особливості, загальновідомі факти, пов’язані події;

– мережева дані: патерни поширення мережею, зв’язки між користувачами.

Таким чином в цій задачі присутні текстові та графічні дані а також більш складні структури, такі як дерева та графи. Центром уваги цього

дослідження є переважно текстові вхідних данні, а саме наповнення контенту та контекст подій і загальновідомих фактів. Тобто вагомою частиною задачі розпізнавання дезінформації є обробка природної мови (англ. Natural language processing, тут і далі NLP).

1.1 Огляд основних інструментів NLP

Кожен алгоритм, що працює з натуральною мовою потребує попередньої обробки тексту, серед основних її етапів можна виділити токенізацію – розбиття на окремі слова чи словосполучення, нормалізацію або лематизацію – обрізання слів до їхнього кореня або спрощення до базової форми, без префіксів, суфіксів тощо), видалення стоп-слів (що, або, це, б, же, is, of тощо), визначення частин мови, власних назв, семантичного зв’язку між словами, вилучення різних ключових характеристик (специфічні слова, емоційні описи, полярність значень тощо) та векторизація – процес перетворення тексту, слів, словосполучень у векторне представлення (масив чисел).

Також корисним терміном тут і далі є поняття n-грами. Це фактично набір токенів по n слів. Наприклад речення «Сьогодні хороша погода.» може бути розбито на такі 2-грами (біграми): «Сьогодні хороша» і «хороша погода».

Важливою компонентою NLP є векторизація тексту, цей інструмент перетворю слова, речення або цілі документи на числові вектори, які зможе розуміти машина. Більш складні та просунуті алгоритми векторизації дозволяють вирахувати числове значення слів враховуючи їх сенс. Фактично таке прогресивне векторне представлення можна зобразити в певному n-вимірному просторі, і чим ближчі два представлення – тим ближчі слова за значенням. Наприклад вектори слів «король» і «королева» мають бути поруч, десь недалеко від них знаходитиметься слово «Англія». Існує багато технік векторизації тексту:

а) статистичні методи;

– One-Hot Encoding (однократне кодування): кожне слово представляється вектором, де лише одна компонента дорівнює 1, а решта – 0. Недоліком є великі розміри векторів і відсутність семантичних зв’язків між словами;

– Bag-of-Words (мішок слів, скорочено BOW): формує вектор за частотою появи кожного слова в документі, ігноруючи порядок слів, фактично це словник з унікальними словами і їхнім лічильником. Недолік: втрачається інформація про синтаксис і контекст;

– TF-IDF (Term Frequency-Inverse Document Frequency): враховує не тільки частоту слова в документі, а й його унікальність у всіх вхідних текстах. Це дозволяє значно зменшити розмір векторів в порівнянні з попередніми варіантами та позбутися впливу слів, що повторюються часто але не несуть значення і натомість надати більшу вагу рідкісним і змістовним словам. Індекс TF виглядає так:

$$TF(t,d) = \frac{N_{t,d}}{\mathrm{N}_d}, \tag{1.1}$$

де $N_{t,d}$ – кількість повторень терміну t в документі d;

N_d – кількість всіх термінів в документі d.

Індекс IDF вираховується за такою формулою:

$$IDF(t,D) = \log\left(\frac{\mathrm{N}}{1+|d \in D : t \in d|}\right), \tag{1.2}$$

де N – кількість документів,

$|d \in D : t \in d|$ – кількість документів, що містять термін t.

Відповідно формула для індексу TF-IDF:

$$TF\text{-}IDF(t,d,D) = TF(t,d) \times IDF(t,D). \tag{1.2}$$

б) методи вбудовування слів (Word Embeddings):

– Word2Vec: використовує моделі Skip-Gram (передбачає слова які часто стоять поруч з заданим) та CBOW (безперервний «мішок слів» – передбачає пропущене слово) для навчання векторних представлень слів так, щоб слова з подібними контекстами мали схожі вектори. На рисунку 1.2 зображено схематичні архітектури цих моделей;

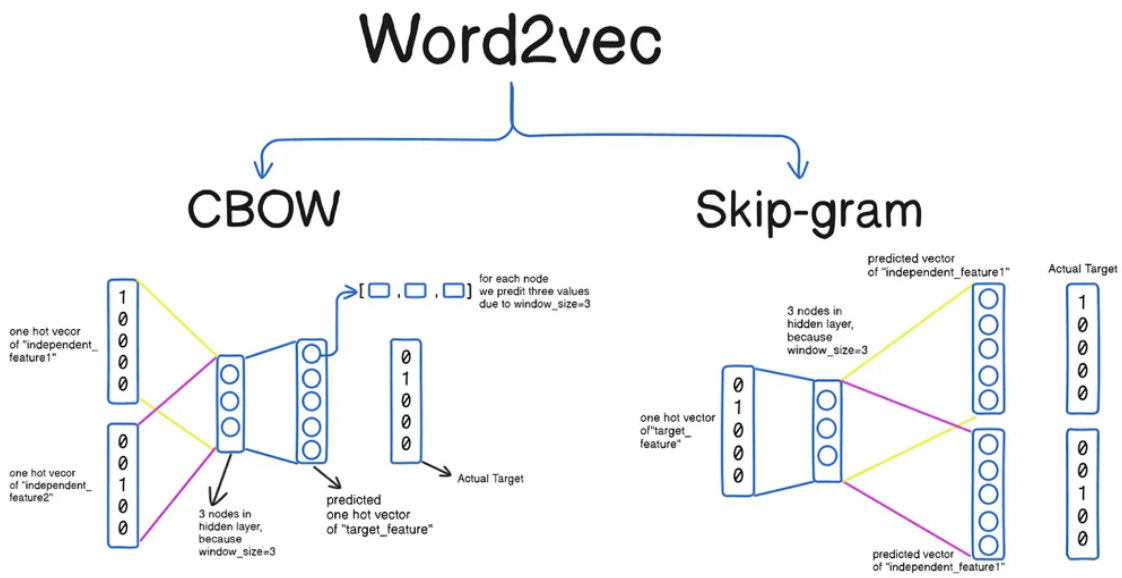

Рисунок 1.2 – Схематичне зображення архітектури моделей, що використовуються для Word2vec [8].

– GloVe (Global Vectors): побудований на матриці, що містить глобальні статистичні дані про спільне уживання слів у корпусі, що дозволяє краще враховувати відносини між словами. Підходить коли набір даних недостатньо великий [9];

– FastText: це модифікація word2vec яка замість BOW для представлення слів використовує n-грами з літер (склади), і це покращує роботу з рідкісними словами і дозволяє враховувати морфологічні особливості [10];

в) контекстуальні моделі: BERT, ELMo, GPT та інші трансформери. Використовуючи механізм само уваги (self-attention) вони генерують контекстно-залежні вектори, де представлення слова залежить від його оточення. Це дозволяє точніше відображати багатозначність і контекстуальні нюанси [11];

г) векторизація документів: doc2vec, Sentence-BERT: створюють вектори для речень або цілих документів, що дозволяє аналізувати тексти як єдине ціле, а не лише окремі слова.

В контексті розпізнавання маніпуляцій в тексті важливо відслідкувати та обрахувати емоцію, що несе в собі та викликає інформація. Базовим рішенням такої проблеми є аналіз загальної тональності тексту (англ. sentiment analysis or opinion mining). Це клас методів аналізу тексту в комп'ютерній лінгвістиці, що призначений для автоматизованого виявлення емоційно забарвленої лексики. Зазвичай за допомогою алгоритмів аналізу тональності текст класифікують як позитивний, негативний або нейтральний, іноді додається певна градація між цими станами. Загалом є три основні групи рішень:

– лексичний підхід (англ. lexicon-based): ґрунтується на заздалегідь визначеному списку слів, позначених їхньою тональною (сентиментальною) полярністю, наприклад, слово «щасливий» позначене як позитивне, «жахливий» – негативне. Кожне слово також може мати кількісне значення тональності. Загальна тональність тексту розраховується шляхом підсумовування або усереднення показників його слів, часто з використанням спеціальних правил для заперечень як от в слові «непогано», або слів, що підвищують інтенсивність, наприклад словосполучення «дуже щасливий». Серед переваг такого методу є простота, легкість інтерпретації та відсутність потреби в даних для навчання. Проте цей підхід вимагає великої бази оцінених слів та може мати проблеми з контекстом, сарказмом або предметно-орієнтованою мовою, якої немає в лексиконі;

– традиційні алгоритми ML: навчання моделі керованого машинного навчання на великому наборі даних текстів, які були вручну позначені як позитивні, негативні або нейтральні. В якості ознак можуть використовуватися BOW, TF-IDF та n-грами, після отримання з тексту вони подаються в класичні алгоритми, такі як Наївний Баєсів класифікатор (англ. Naive Bayes), опорні вектори (англ. Support Vector Machine – SVM), логістична регресія (англ. Logistic Regression) або Випадковий ліс (англ. Random Forest). Такий підхід допомагає вивчати складні закономірності з даних, часто точніші, ніж прості лексичні методи. Проте вимагає значного обсягу розмічених даних для навчання і створення ознак не завжди ефективне;

– алгоритми глибинного навчання (англ. DL): підхід використовує нейронні мережі для безпосереднього вивчення тональності з необробленого тексту. Векторні представлення слів фіксують їхнє семантичне значення та передаються до CNN (convolutional neural network – згорткова мережа), RNN (recurrent neural network – рекурентна нейронна мережа) чи LSTM (long-short-term memory – довго-коротко-строкова пам’ять) або трансформера. Мережа вивчає контекстуальні зв'язки. Попередньо навчені моделі часто доналаштовують на конкретних наборах даних для аналізу тональності. Цей підхід дає найкращу точність, відмінно справляється з уловлюванням контексту та нюансів (сарказм, іронія), може обробляти великі та різноманітні набори даних. Однак великим недоліком є значний обсяг обчислювальних ресурсів для навчання з нуля (хоча дотренування є менш виснажливим). Також ця група алгоритмів дає менш прозорий і зрозумілий результа;

– гібридні алгоритми: поєднують в собі лексичний підхід та ML або DL алгоритми.

Визначення більш складної емоційної характеристики працює аналогічно, проте в якості класів мають використовуватися замість тональної забарвленості конкретні емоції, такі як радість, сум,

розчарування, гнів тощо. В такому випадку важливо підібрати максимально ефективну класифікацію.

1.2 Аналіз існуючих рішень

В рамках задачі розпізнавання неправдивої інформації існує широке різноманіття підходів для її вирішення. Основними відмінностями є дані, що використовуються для аналізу, спосіб навчання (з вчителем чи без) а також алгоритми, що використовуються для обробки даних. Ієрархічна структура не дуже доцільна для опису існуючих методик розпізнавання маніпуляцій в тексті, тому на рисунку 1.3 зображено більш зручне представлення основних підходів для вирішення задачі розпізнавання брехні в медіа контенті, запропоноване в цьому досліджені.

Рішення, що ґрунтуються на алгоритмах навчання з вчителем вимагають повністю помічених даних. Такі роботи переважають серед досліджень і є найваріативнішими. Вони показують хороші результати, є зрозумілими. Єдиним недоліком є потреба у великих достовірно помічених наборах даних.

Частково кероване навчання має на увазі що є частина даних з відмітками та частина без. Серед досліджень в обраній темі цей напрямок стає дуже популярним протягом останніх років, це пов’язано переважно з потребою тренування на нових великих непомічених даних та зростанні популярності та якості генеративних мереж, особливо змагальних (англ. generative adversarial network – GAN). Такі рішення подають великі надії і непогано працюють в умовах неякісних, неповних та розмитих даних.

У випадках, коли відсутні будь-які мітки на даних, наприклад при аналізі мережевого розповсюдження використовуються алгоритми навчання без вчителя, і досліджень в цій методиці також стає більше. Через відсутність будь-яких лейблів такі алгоритми працюють зовсім інакше й здатні знаходити цікаві патерни та закономірності в інформації, що

аналізуються. Існують дослідження де такі алгоритми використовуються для виявлення неправдивої новини на ранніх етапах її поширення, коли може бути недостатньо інформації.

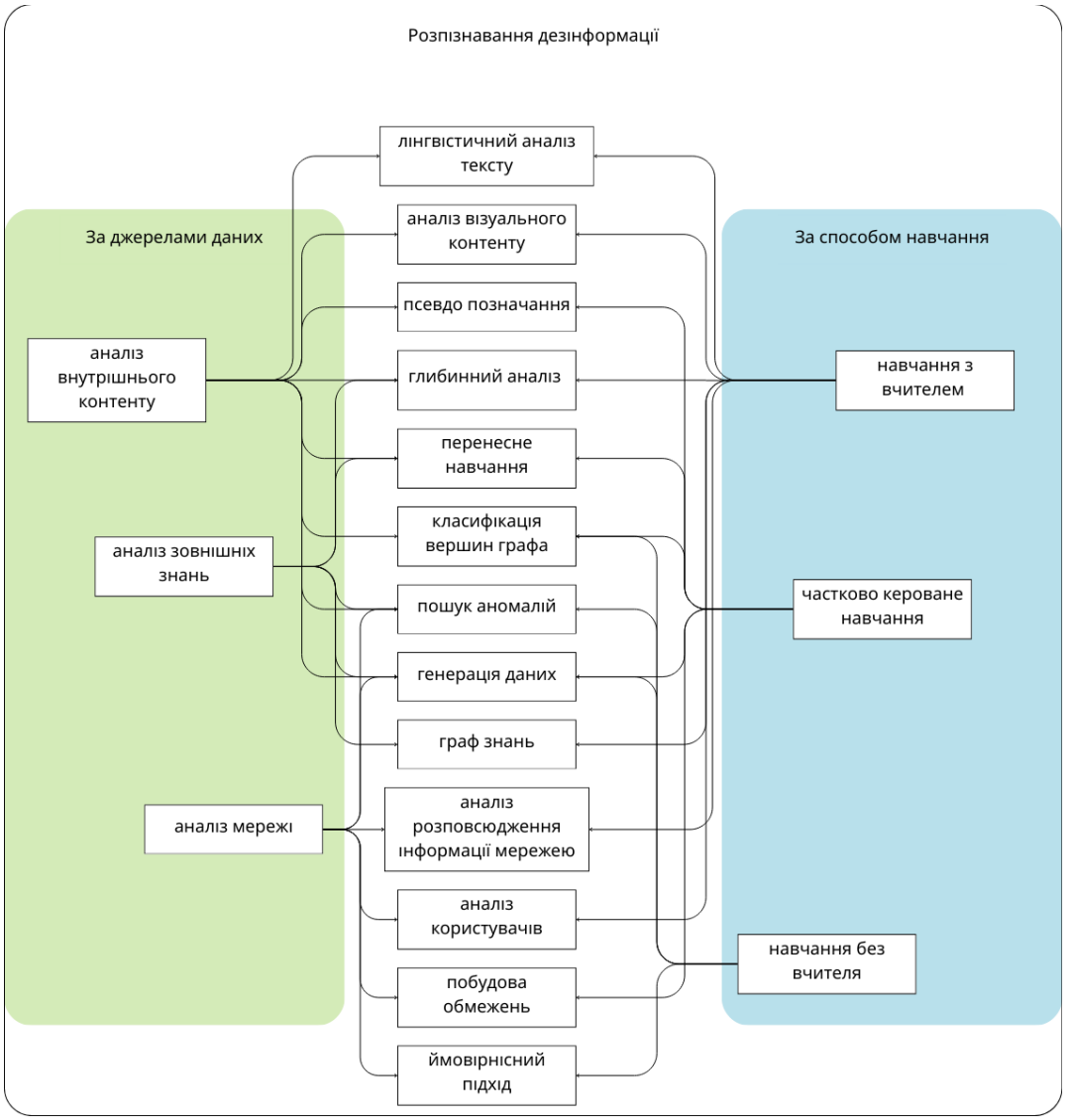

Рисунок 1.3 – Запропонована класифікація основних алгоритмічних підходів для вирішення задачі розпізнавання дезінформації

Рішення, що для аналізу використовують внутрішні дані, працюють переважно з обробкою тексту, іноді також зображень, іноді лише зображення. Такі алгоритми не враховують зовнішніх даних, тому не здатні оцінити дійсність інформації, проте дуже ефективно здатні знаходити

закономірності в стилістиці висловлення думки які вказують на високу ймовірність брехні. Також найкраще працюють для розпізнавання гумористичних текстів.

Використання зовнішньої інформації доповнює попередній підхід і значно покращує можливості розпізнавання. Загальновизнані факти, інформація про об’єкти і суб’єкти в тексті, про автора та про контекст подій дає можливість оцінити фактичну правдивість інформації незалежно від стилю її подання.

Алгоритми, що ґрунтуються на використанні мережевих даних демонструють зовсім інший підхід, замість аналізу того, що міститься власне у тексті обробляють інформацію поширення новини, і того як різні користувачі реагують і впливають на такі новини. В першу чергу це дозволяє набагато краще оцінити ефект фейкової інформації в інтернеті на соціальному рівні, але також такий підхід ефективно показує себе в ситуація де є нестача контекстної інформації. Проте такі рішення потребують великий наборів даних, які важко дістати, або які захищені користувацькими правами.

1.2.1 Аналіз внутрішнього контенту

Найпростішими алгоритмами цієї групи є лінгвістичні моделі, що базуються на використанні граматичних, лексичних і тональних характеристик, словесних шаблонах, та простих алгоритмах векторизації тексту. Вони добре працюють для визначення стилістики та емоційної забарвленості тексту, дозволяють відслідкувати прості закономірності притаманні брехні, можуть бути використані для розпізнавання сарказму, іронії тощо. Ці алгоритми є простими, швидкими та не потребують дуже великих ресурсів при навчанні.

Чудовим прикладом є робота 2017 року «Automatic Detection of Fake News»[7], в якості вхідних характеристик бралися 2- і 3- грами з BOW, їхні

TF-IDF представлення, пунктуація, ознаки читабельності, такі як кількість символів, довжина та складність слів, деякі метрики читабельності та психолінгвістична складова тексту зібрана за допомогою LIWS (Linguistic Inquiry and Word Count – лінгвістичне дослідження та кількість слів) – дуже популярного інструменту вилучення емоційної забарвленості, що має дуже глибокий набір позначень. Для тренування використовувалась лінійна модель опорних векторів (бінарна класифікація), набір даних був зібраний з різних джерел, був збалансованим та містив правдиві та фейкові новини в 6 темах. В середньому кращі результати дало поєднання всіх характеристик та окремо читабельності, зі значеннями метрики F1 0.74 та 0.79 відповідно.

Якщо говорити про класифікацію між звичайним текстом і таким що є саркастичним, то прикладом може бути робота «Implementation of Emotional Features on Satire Detection» (2018) [12]. Тут були використані такі ознаки як біграми та триграми слів в BOW, емоційні риси (EmoLex) вилучені за допомогою двигуна аналізу настроїв та соціального пізнання SEANCE, полярність емоцій цього ж двигуна, мішок емоцій (BOSE) і його TF-IDF представлення. Для тренування було обрано два варіанти: SVM та Random forest. Останній показав вищі результати, найкращими характеристиками є поєднання всіх для наборів даних зібраних з новин та соціальної мережі Twitter, а також чистий BOW для відгуків на сайті зі значеннями F-міри 0.78 та 0.82 відповідно.

Проте лінгвістичний підхід має значний недолік через ігнорування контексту і відсутність дійсної перевірки знань, дуже добре цю проблему описано в дослідженні «Fake news detection via NLP is vulnerable to adversarial attacks»[13]. В цьому документі за приклад лінгвістичної моделі береться модель платформи «FakeBox», і для демонстрації недоліків такого підходу було створено спеціальні екземпляри текстових новин, з використанням таких технік: викривлення фактів, зміна місцями об’єкту та суб’єкту, логічну плутанину. В результаті тесту такими екземплярами незважаючи на очевидну дезінформацію модель класифікувала новини як

більш схожі до правди. Натомість якщо подати на вхід правдивий текст написаний не в журналістському стилі, він з більшою ймовірністю буде визначений як фейковий. Ці тести підкреслюють дві основні проблеми такого підходу: ігнорування відвертої брехні, яка написана гарно та упередженість до стилістики написання тексту.

Використання нейронних мереж для вилучання характеристик з тексту значно покращує ефективність розпізнавання, оскільки здатні зрозуміти контекст та більш глибинні закономірності.

Показовою роботою з таким підходом є «A Unified Training Process for Fake News Detection based on Fine-Tuned BERT Model» [14]. Вона є доволі показовою, оскільки розроблена там модель досягла значення метрики F1 = 0,97. Тут було представлено особливий підхід до попередньої обробки тексту, а саме видалення слів, коротших за 3 літери, це призвело до пришвидшення роботи алгоритму (з 22 годин до 12). Також цікавим був підхід поєднання трьох датасетів: три екземпляри однієї архітектури мережі на основі BERT були натреновані на 3-х відповідних датасетах, і потім дотреновані разом.

В контексті розпізнавання сатири в тексті теж застосовують глибинний підхід, наприклад в роботі «Context-Driven Satire Detection With Deep Learning» (2022) [15]. Для входу було виділено багато різноманітних характеристик: Juvenalian Feature (базується на частоті слів властивих сатирі), Horatian Feature (подібно до попереднього, але інші тональності сатири), Menippean Feature (слова пов’язані з вірою, культурою, політикою), Joy Feature (частота слів пов’язаних з радістю), бінарна характеристика присутності чи відсутності негативних слів, ознаки зібрані за допомогою вищезгаданої лексичної моделі LIWS та CNN. Логістична регресія показала найкращу F1 оцінку = 0.87.

1.2.2 Аналіз зовнішніх даних

Дуже поширеним способом представлення інформації з використанням зовнішніх даних є граф знань (ГЗ), структура, що зберігає об’єкти як вершини та зв’язки між ними як ребра. ГЗ дозволяють логічно поєднати різних типів з численних джерел, зберігши їх взаємини, а також легко вбудувати векторизацію даних та їхнє порівняння. Важливим питанням методів, що базуються на аналізі зовнішніх даних є методика отримання додаткової інформації про зміст новин й об'єднати її з внутрішніми знаннями. В цьому підході дуже відомою є робота «Meet The Truth: Leverage Objective Facts and Subjective Views for Interpretable Rumor Detection» 2021 року [16], де була презентована модель LOSIRD (використання об’єктивних фактів та суб’єктивних поглядів для виявлення чуток, які можна інтерпретувати). Дослідження зосереджене на гібридній моделі виявлення чуток, яка поєднує пошук доказів і та прогнозування чуток, використовуючи Вікіпедію як джерело інформації. Вона оцінює ефективність знайдених доказів у спростуванні чуток, показуючи, що лише близько 14,8% доказів були релевантними. Рішення включає в себе векторизацію слів, перетворення їх у векторизовані речення за допомогою BiLSTM (англ. Bidirectional Long Short-Term Memory – двонаправлена довго- коротко- строкова пам’ять), перетворення цих даних на граф і його вилучення ознак за допомогою моделі GraphSAGE і простий класифікатор. В результаті тестування модель показала значення метрики F1 вище 90% для всіх наборів даних.

Варто також згадати роботу «Fake News Detection Through Graph-based Neural Networks: A Survey» [17], ключовими моментами роботи було побудова графу знань на вже відомих речах (внутрішні знання) та порівняння їх з зовнішніми знаннями для подальшого виявлення невідповідностей, а також використання графів поширення. Основою були

GCN та CNN архітектури. Найкращі результати серед алгоритмів ґрунтованих на перевірці знань показала вищезгадана модель LOSIRD.

В досліджені «Adversarial Active Learning based Heterogeneous Graph Neural Network for Fake News Detection» 2021 року [18] використовується гетерогенний граф для обробки інформації, що містить вузли авторів, новин і тематик. Граф обробляється HGAT класифікатором і HGAT селектором. HGAT (англ. Hierarchical Graph Attention Network) – це модель, яка використовує ієрархічний механізм уваги для обробки гетерогенних графів, складається з двох компонентів, один з яких обробляє вершину локально й другий аналізує глобально всі типи вершин. Рішення показало перевагу в точності та F1-оцінці в порівнянні з іншими методами, навіть при використанні лише 20% навчальних даних.

ГЗ запропонований в дослідженні «KAN: Knowledge-aware Attention Network for Fake News Detection» 2021 року [19] відрізняється за структурою, оскільки в якості вузлів використовує різні сутності, такі як люди, місця, події тощо замість новин. Для обробки графа використовується нова модель KAN (англ. Knowledge-aware Attention Network – обізнана мережа уваги). Її особливість у використанні двох механізмів уваги, один направлений на оцінку релевантності сутності до новини та другий додає до оцінки контекст. Результати тестування показують, що модель KAN перевершує існуючі методи за всіма основними метриками, зокрема метрика F1 досягає 0.8435 на наборі даних PHEME, що є вищим, ніж у інших моделей.

Ще одне дослідження 2021 року «Compare to The Knowledge: Graph Neural Fake News Detection with External Knowledge» [20] запропонувало модель CompareNet, яка також працює з гетерогенним графом, в якому вузлами є речення, теми та сутності, тут також в користовується HGAT, проте основною особливістю є спеціальний алгоритм порівняння сутностей. При тестуванні модель обійшла попередників у метриці F1 в середньому на 3%.

Інший приклад використання ГЗ це робота «Incorporating Relational Knowledge in Explainable Fake News Detection» 2022 року [21], де вузлами графу є люди, місця та організації, і велика увага приділяється надійності ребер (відносин). Сама ж модель KGF (англ. knowledge graph enhanced framework – фреймворк посилений графом знань) використовує спільну увагу а також приймає деякі додаткові ознаки, такі як коментарі користувачів. Чітка структура графу дозволяє моделі реалізовувати механізм пояснення. Він працює шляхом використання уваги для виділення важливих ознак у вхідних даних, таких як слова або фрази, що впливають на прийняття рішення моделі, аналізує, які частини інформації найбільше сприяють результату, надаючи прозорість у процесі прийняття рішень. Однак, цей механізм має обмеження, оскільки зосереджується лише на існуючих ознаках і не завжди враховує реляційні залежності між ними. KGF покращує точність і F1-оцінку на 17% у порівнянні з моделлю логістичної регресії на Celebrity та на 30% у точності і 35% у F1 на PolitiFact.

У задачі виявлення дезінформації англомовні джерела часто мають багато анотованих даних на відміну від інших, менш поширених мов, мають мало даних. Тому корисним може буде перенесне (трансферне) навчання. Його мета полягає в отриманні узагальнених знань шляхом використання багатих на анотації даних з вихідної області та перенесення цих узагальнених знань на цільове завдання, що допомагає в навчанні цільового завдання. Тобто в контексті різних мов, модель навчається на англомовному наборі даних, щоб отримати узагальнені попередні знання, і далі переноситься на завдання виявлення фейкових новин іншими мовами. Наприклад робота 2021 року «Cross-lingual COVID-19 Fake News Detection» [22] використовує цей підхід для розпізнавання дезінформації китайською, хоча сама модель BERT навчалася на наборі даних, що складається з новин англійською про COVID-19 з трьох джерел: ReCOVery, FakeCovid та CoAID. Тестування проводилось на зібраних китайських новинах, серед яких 86 були фейкові та 114 правдивими. Набір даних

включав метадані, такі як заголовок, автор та час публікації, але основна увага приділялася тексту новин.

Ще одним інструментом вирішення задачі розпізнавання дезінформації при недостатній кількості розмічених навчальних даних є GAN. Робота «A Fuzzy Detection System for Rumors through Explainable Adaptive Learning» 2021 року [23] презентує модель Graph-GAN, яка працює на основі графового вбудовування та генеративного змагального навчання. Модель створює детальні простори ознак через графове кодування та використовує безперервне змагальне навчання між генератором і дискримінатором для розпізнавання дезінформації в умовах відсутності міток. Це дозволяє покращити точність виявлення брехні, використовуючи адаптивну оптимізацію та глибше розуміння знань. Graph-GAN продемонстрував найкращі результати серед усіх протестованих в дослідженні моделей у точності на наборі даних Twitter. Однак результати на наборі даних Weibo виявилися гіршими, що може бути пов'язано з особливостями обробки китайських текстів.

Варто також зазначити, що для задачі розпізнавання дезінформації добре підходять великі мовні моделі (LLM), такі як GPT. Вони навчанні на величезній кількості фактів, добре розуміють людську мову, а тому здатні перевірити подану їм інформацію з правильним запитом і надати обґрунтування. У випадку нестачі контексту, його можна передати додатковим модулем пошуку в мережі чи в додатковій базі даних. Проте такі моделі потребують колосальних навчальних даних.

1.2.3 Аналіз мережі

Мережевий підхід має на увазі оцінку додаткової зовнішньої інформації яка на пряму впливає на поширення новини в інтернеті, переважно це дані що стосуються або користувачів, або патернів розповсюдження мережею. Алгоритми що аналізують інформацію про

користувачів часто базуються на оцінці надійності акаунтів, зв’язках між людьми та на їхні коментарі, проте насправді такі рішення є переважно гібридними, оскільки також обробляють текст самої новини. Наприклад у роботі «MM-COVID: A Multilingual and Multimodal Data Repository for Combating COVID-19 Disinformation» 2020 року [24] представлено багатомовний (6 мов) і багатовимірний набір даних MM-COVID, спрямований на виявлення фейкових новин, пов'язаних з COVID-19. У ньому детально описано процес збору даних, який включає в себе збір новинного контенту, соціальну активність у мережі Twitter та перевірку фактів з надійних джерел. Дані з Twitter використовуються для збору соціальних взаємодій, таких як твіти, відповіді та ретвіти, що стосуються новинного контенту. Для цього формуються пошукові запити на основі URL, заголовка та першого речення джерела, а потім використовуються API Twitter для збору відповідних постів. Зібрані дані допомагають аналізувати емоції користувачів їхні зв’язки та виявляти патерни, пов'язані з поширенням фейкових новин. Для подальшого аналізу використовувався фреймворк dEFEND (англ. Explainable Fake News Detection – пояснювальне розпізнавання фейкових новин), який був представлений в роботі «dEFEND: Explainable Fake News Detection» 2019 року [25]. Структура dEFEND включає чотири основні компоненти: кодувальник новинного контенту, кодувальник коментарів користувачів, компонент спільної уваги між реченнями новин і коментарями, та компонент прогнозування фейкових новин. Ця робота перевершує своїх попередників в цьому підході, такі як CSI (англ. Content-Source Interaction – взаємодія між контентом і джерелом) та TCNN-URG (англ. Two-level Convolutional Neural Network for User Response Generation – дворівнева згорткова нейронна мережа для генерації відповідей користувачів). Міра F1 на обох використаних наборах даних (PolitiFact і GossipCop) в середньому вища на 5.33%. Крім того здатність до пояснювальності вища на 30%, оскільки може ранкувати коментарі, що пояснюють краще вище, ніж не пояснювальні.

Також вплив емоцій в тексті та в реакціях користувачів на розпізнавання неправдивої інформації було досліджено в роботі «Mining Dual Emotion for Fake News Detection» 2021 року [26]. Автори запропонували новий набір ознак, названий подвійні емоційні ознаки, що представляє та емоційне забарвлення власне контенту, коментарів та їхній взаємозв’язок. Ці ознаки були легко інтегровані в існуючі системи виявлення фейкових новин, використовувалися моделі BiGRU, BERT та HSA-BLSTM, де остання показала себе найкраще й обійшла попередників у показниу F1 на 6%.

Дослідження 2019 року «User-Characteristic Enhanced Model for Fake News Detection in Social Media» [27] зосереджується на вивченні характеристик користувачів та їхніх взаємозв'язків у контексті поширення новин. Дослідники створили гетерогенну мережу, де вузлами є автори, новини та теми, а також використовували моделі, такі як RNN і CNN, для вивчення глобальних і локальних змін характеристик користувачів у процесі поширення. Крім того, було розроблено модель на основі дерева поширення, що дозволяє порівнювати подібність структур чуток для кращого розрізнення різних типів брехні. В результаті було запропоновано систему UCEM (англ. user-characteristic enhanced model – покращена модель з користувацькими характеристиками), яка перевищила попередні значення метрики F1 на 2–4%

Робота «A Graph Convolutional Encoder and Decoder Model for Rumor Detection» 2020 року [28] також досліджує патерни розповсюдження фейкових новин для їх розпізнавання. Основою рішення є граф, де вузли представляють твіти, а ребра – відносини між ними (ретвіти та відповіді). Такі мережі далі оброблялися GCN (англ. graph convolutional networks – графова згорткова мережа), VAE (англ. variational autoencoder – варіаційний автокодувальник) та GAE (англ. graph autoencoder – графовий автокодувальник). В результаті було виявлено, що структура чуток формується динамічно через активне поширення, тоді як структура

правдивих новин є статичною, також чутки мають специфічні властивості зв'язків між вузлами, що відрізняються від зв'язків у не чутках. Найкраще себе проявили мережі з використанням автокодувальників досягнувши значення міри F1 = 0.94.

Показовим прикладом використання мережевих даних з недостатньої кількістю позначок (частково кероване навчання) є дослідження «Weak Supervision for Fake News Detection via Reinforcement Learning» 2019 року [29], де пропонується WeFEND модель. Це рішення покращує виявлення фейкових новин шляхом використання трьох основних компонентів: анотатора, детектора фейкових новин та підсилювального селектора. Анотаратор автоматично присвоює слабкі мітки для неанотованих новин на основі відгуків користувачів, а селектор вибирає високоякісні зразки для навчання детектора. Це дозволяє зменшити вплив шумових міток і покращити точність виявлення фейкових новин. Дослідження також з’ясувало, що розподіл новин змінюється з часом, оскільки новини, що з’являються в різні часові вікна, мають різні характеристики та ознаки. При тестуванні було досягнуто точності 0.824.

Прикладом підходу, що використовує навчання без вчителя є робота «Unsupervised Fake News Detection on Social Media: A Generative Approach» 2019 року [30]. Тут використовується ймовірнісний графовий підхід, що розглядає проблему виявлення фейкових новин як ймовірнісну задачу. В цій роботі правдивість новин та надійність користувачів сприймаються як приховані змінні, вважаючи, що правдивість новин більше пов'язана з надійністю користувачів. Запропоноване рішення використовує алгоритм UFD (англ. Unsupervised Fake News Detection – некероване розпізнавання фейкових новин), використовує ймовірнісну графічну модель для аналізу взаємодії користувачів у соціальних мережах, щоб оцінити правдивість новин та надійність користувачів. UFD дозволяє виявляти фейкові новини, спираючись на ненадійні дані про соціальну взаємодію. Результати дослідження показали, що запропонований алгоритм досягає найкращих

показників точності на наборі даних LIAR, перевершуючи другий найкращий алгоритм на 18.4%. На наборі даних BuzzFeed UFD демонструє найкращі результати, для правдивих новин. Загалом, включення другого рівня взаємодії користувачів значно покращує ефективність виявлення фейкових новин.

Згідно з принципами динаміки соціальної комунікації поведінка користувачів при публікації брехні відрізняється від поведінки при публікації справжніх фактів. Щоб використати ці відмінності для виявлення чуток, деякі дослідники розглядають публікації чуток як аномалії в соціальних мережах і представляють задачу виявлення дезінформації як виявлення аномалій. Наприклад в дослідженні «Unsupervised rumor detection based on users' behaviors using neural networks» 2018 року [31] правдивість новин оцінювалась на основі поведінки користувачів при публікації. Запропоноване рішення використовує RNN для аналізу коментарів і публікацій протягом часу і далі передає ці дані в VAE для визначення аномалій (брехні). Модель досягла точності в 92.49 % та значення метрики F1 89.16%.

1.3 Висновок

Таким чином проблема розпізнавання дезінформації в медіа контенті є актуальною, й містить велику варіативність даних для обробки. Невід'ємною частиною будь-якого рішення цієї задачі включатиме в себе алгоритми NLP. Існує три основних підходи для розпізнавання неправдивої інформації в інтернеті: аналіз внутрішнього контенту, а саме текст і заголовок статті, аналіз зовнішніх даних, таких як контекст подій, автор, загальновідомі факти, та аналіз мережі, що включає в себе глибоке вивчення патернів поширення новин та взаємозв'язки між авторами та коментаторами. Для тренування моделей найчастіше використовується навчання з вчителем, проте багато досліджень також використовують

частково кероване та не кероване навчання. В таблиці Б.1 зібрано короткий огляд існуючих методів.

Найкращою структурою даних для обробки даних в задачі розпізнавання дезінформації є граф знань в різних варіаціях. Серед найбільш ефективних інструментів можна назвати контекстуальну векторизацію, емоційну та стилістичну оцінку тексту, заголовку та коментарів, використання комбінації зовнішніх та мережевих даних, GNN та особливо GAT. Головною спільною проблемою існуючих рішень можна назвати прив’язаність навчальних даних до англійської мови та специфічних тем. Також існує широкий простір для об’єднання та комбінації існуючих інструментів в одну складну та ефективну систему.

2 ПРОЄКТУВАННЯ РІШЕННЯ

2.1 Проєктування бази знань

2.1.1 Огляд існуючих наборів даних

Існує велика кількість наборів даних з фейковими новинами, вони досить різноманітні, містять різні метадані, ознаки, відрізняються за тематиками та мають різні системи позначень. Найбільші й найповніші дані зазвичай публікуються разом з відповідними їм науковими роботами, проте більшість з них є недоступними для загалу, або застарілими, й тому не будуть розглядатися в цьому дослідженні.

Набір даних з новин BuzzFeedNews [32] містить повну вибірку новин, опублікованих у мережі Facebook дев’ятьма інформаційними агентствами протягом тижня напередодні виборів у США 2016 року (з 19 по 27 вересня). Кожен пост і стаття, на яку він посилається, були перевірені п’ятьма журналістами агенції BuzzFeed. Всього датасет містить 1 627 статей, до кожної є наступний набір ознак : заголовок, текст, автор, організація, посилання, політична направленість (праве / ліве крило чи нейтральна) та власне клас. Представлено 4 класи: «переважно правда», «переважно брехня», «суміш брехні та правди» а також «відсутність фактичного вмісту».

Іншим популярним набором даних є набір LIAR, який було презентовано разом з вже згаданим вище науковим дослідженням [6]. Тут представлено 12 836 новин, які були зібрані з сайту американського інформаційного агентства PolitiFact, що займається розвінчанням брехні в інтернеті. Дані позначенні одним з сіми класів: «правда», «переважно правда», «правда наполовину», «частково правда», «трохи правда», «брехня» і «штани у вогні»; останній клас посилається на англійську приказку і позначає найбільший ступінь брехні. Кожна новина представлена

такими ознаками: заголовок, текст, автор, посада автора, політична партія, штат, предмет обговорення, контекст місця чи події, та кількість статей позначених кожним з класів.

Ще одним набором даних, який використовує викриття новин організацією PolitiFact є Politifact Fact Check [33]. Він містить аналогічні до попереднього позначки класів, та має сім ознак, серед яких: автор заяви, заголовок заяви, дата заяви, її джерело, автор перевірки, дата перевірки, та посилання на статтю перевірки. Всього набір містить 21 152 записи, що покривають 2008–2022 роки.

Одним з дуже популярних датасетів, що містить дані про поширення інформації є набір Twitter15 та Twitter16 [34], він був опублікований в 2017 році, записи позначені чотирма класами: «правдива плітка», «брехлива плітка», «не плітка» та «не визначено». Набір даних бере інформацію з мережі Twitter, тому кожен екземпляр містить ідентифікатор посту, заголовок, а також дерево поширення, де відображається взаємозв’язок відповідей та репостів.

Набір данихPHEME [35] побудований інакше, він містить ряд подій, і до кожної з них перелік правдивих постів і брехні щодо цієї події (два класи). Таким чином. Остання версія цього набору даних (2018 рік) містить 9 подій і 137 496 постів. Серед ознак присутня тема (подія), заголовок посту, його текст, посилання відповіді (поширення), дата, автор і купа інших метаданих.

Набір даних Fakeddit 2019 року [36] в якості інформації використовує пости мережі Reddit, серед ознак представлено заголовок, зображення, дату, домен, рейтинг, відповідь, автор, деякі інші метадані. Особливістю цього датасету є наявність трьох груп класифікації, а саме група з двох класів (правда, брехня), група з трьох класів (правда, брехня, брехня що містить правду) та група з шести класів: маніпулятивний контент, правда, сатира, хибний зв’язок, оманливий контент, повна брехня.

Крім політичних статей велику частину наборів даних з дезінформацією представляють датасети з медичними фейками, особливо новини, пов’язані з вірусом COVID-19. Наприклад набір даних FakeHealth [37] бере інформацію з мережі Twitter і містить заголовок статті, текст, посилання, ключові слова, теги, дата, автор, зображення, відповіді, інформація про користувача і додаткові метадані.

CoAID [38] – це набір даних про дезінформацію щодо COVID-19 у сфері охорони здоров'я. Всього тут зібрано 5 216 новин протягом чотирьох часових проміжків, зокрема фейкові новини з веб-сайтах і соціальних платформ. Основні ознаки в цьому наборі даних це заголовок, посилання, ідентифікатор відповідника в мережі Twitter, якщо такий є, інформація про відповіді користувачів.

Набір даних FakeCovid [39] є одним з небагатьох багатомовних наборів даних, який містить 7623 перевірених новинних статей про COVID-19, зібраних протягом 2020 з 92-х сайтів, що займаються перевіркою фактів. Данні розподілені за 11 категоріями, проте не дуже чіткими, основними можна виділити «брехня», «часткова брехня», «оманлива інформація», «відсутність доказовості» та «невідомо». Набір містить дані 40-ма мовами зі 105 країн світу, Україна та українська мова серед них, проте в дуже маленькій кількості. Серед ознак є заголовок, текст, категорія, дата, джерело, посилання, посилання на спростування, країна, мова, і ще деякі метадані.

Датасет ReCOVery [40] містить 2029 статей пов’язаних з COVID-19 з різних джерел різних країн (переважно США), деякі з публікацій мають відповідник у мережі Twitter. Представлено два класи (надійний, ненадійний) та наступні ознаки: посилання, зображення, заголовок, текст, автор, країна, політична упередженість, ідентифікатор наявного поста в мережі Twitter.

Ще один багатомовний датасет в цій галузі це MM-COVID [41], мов представлено 6: Англійська, Французька, Італійська, Іспанська, Хінді та

Португальська. Зібрано широкий перелік ознак, серед яких автор, посилання, джерело, заголовок, деякі метадані, інформація з Twitter тощо. Дані мають два класи: правда або брехня. Особливістю цього датасету є наявність скрапера, коду, який допомагає у діставанні новин.

Набори даних для розпізнавання сатири не потребують такого рівня перевірки достовірності, як для перевірки на дезінформацію, тому для отримання більш актуальних даних було обрано шукати подібні набори на платформі Kaggle. Для огляду було обрано три найкращих. Серед них датасет News Headlines Dataset For Sarcasm Detection [42], який бере інформацію з платформ The Onion та HuffPost. В наборі представлено два класи і дві ознаки: заголовок і посилання. Сет даних Tweets with Sarcasm and Irony містить комедійні пости з Twitter поділені на чотири класи: сарказм, іронія, звичайний текст і фігуративна мова.

У табличці Б.2 зібрано загальний огляд описаних вище доступних наборів даних для розпізнавання різних видів не правдивої інформації.

2.1.1 Додаткові джерела інформації

Для більш глибокого аналізу необхідні додаткові джерела інформації, це мають бути ресурси з загально доведеними фактами а також публічні платформи перевірки інформації. Перші дозволять розумітися на загальних речах, а другі можуть надати додатковий контекст або вже розвінчану брехню.

Серед платформ, що займаються перевіркою фактів дуже популярною є американська журналістська організація PolitiFact. Тут публікують спростування заяв популярних спікерів, спеціалісти проводять ретельну перевірку та оцінюють інформацію за одним з 7 класів. Серед переваг є варіативність джерел які перевіряються та персоналій. Всі статті знаходяться в навколо політичній тематиці, проте в цих межах є доволі варіативними. Цю платформу можна використовувати як джерело

перевіреного контексту та репутації автора. Платформа не має публічного API, проте існують бібліотеки для зручного зчитування її сторінок. Недоліком цієї платформи є виключно американський контекст, там можна знайти перевірку новин про Україну, проте це американські новини від американських діячів. На рисунку 2.1 зображено приклад перевірки факту на платформі PolitiFact.

Рисунок 2.1. – Приклад перевірки факту на платформі PolitiFact.

Більш універсальним є сайт від компанії Google FactCheck. Він містить звіти-оцінки різних відносно перевірених користувачів на публічні заяви. Кожен подібний звіт представляє собою окрему сторінку якоїсь людини чи організації зі статтею, що оцінює вірогідність якогось тексту, картинки, новини тощо. Така оцінка має містити автора звіту, оцінку в текстовому та числовому вигляді разом з її можливими межами, посилання на звіт, посилання на новину, що оглядається, на її автора тощо, ключові слова. Варіативність таких оглядів є дуже високою, в тому числі тут можна знайти велику кількість новинних звітів українською. Платформа має публічне API, що полегшує доступ, проте надійність може бути трохи нижча за попередньо оглянуту платформу, а також в більшості випадків звіти роблять саме про дезінформацію, рідко зустрічаються огляди з оцінкою

«правда». Оціночна шкала між звітами відрізняється через різних авторів. На рисунку 2.2 зображено приклад звітів на платформі FactCheck.

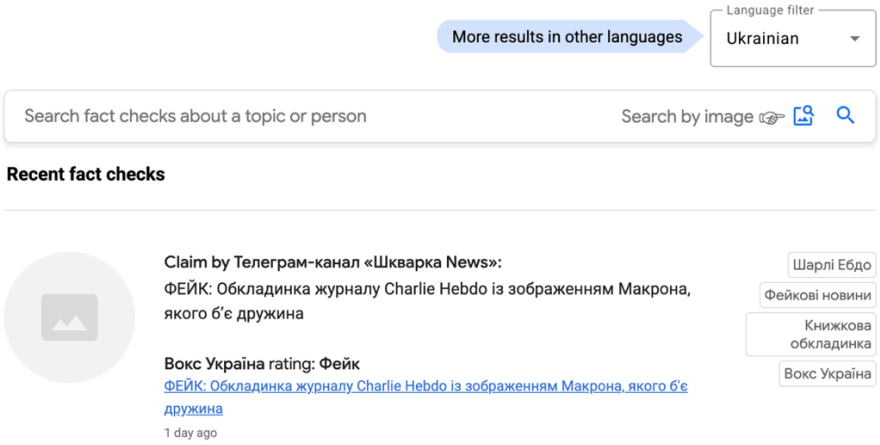

Рисунок 2.2 – Приклад звіту про фейкову новину на платформі FactCheck

Ще одним джерелом перевіреної брехні може бути українська журналістська платформа «Ґвара медіа». Вони мають окремий відділ з перевірки новин, які відправляються стурбованими користувачами. Зазвичай перевірені новини поділяються на два класи: фейк та маніпуляція, ці значення містяться першим словом в заголовку. Це джерело можна використати в якості навчальних даних. На рисунку 2.3 Зображено приклад заголовків на платформі «Ґвара медіа».

Хорошим джерелом загальної інформації є відома база даних Google Knowledge Graph. Вона містить дані про різні події, місця, дати, предмети, науковців, політиків та відомих людей тощо. Дані в базі постійно оновлюються, в якості джерел використовуються авторитетні платформи. При запиті до такої бази можна шукати і отримати відповідь українською мовою. Ця база даних має API, що полегшує отримання необхідної інформації, проте доступ до нього є обмеженим, а саме необхідно мати ключ авторизації від хмарних застосунків Google, після деяких налаштувань такий ключ можна отримати безкоштовно на пробний період. На рисунку 2.4 зображено приклад запиту до Google Knowledge Graph API.

Рисунок 2.3 – Приклад звітів про дезінформацію двох різних видів на українській платформі «Гвара медіа»

Іншим відомим джерелом є сайт Wikipedia. Це дійсно величезна база знань, вона є повністю безпечна, проте через свою природу розподіленої між користувачами системи знань, не всі дані можуть бути достовірними, тож це варто враховувати. Wikipedia також має своє API та його програмні реалізації мовою програмування python. Для пошуку доступна українська мова. На рисунку 2.5 зображено приклад запиту до Wikipedia.

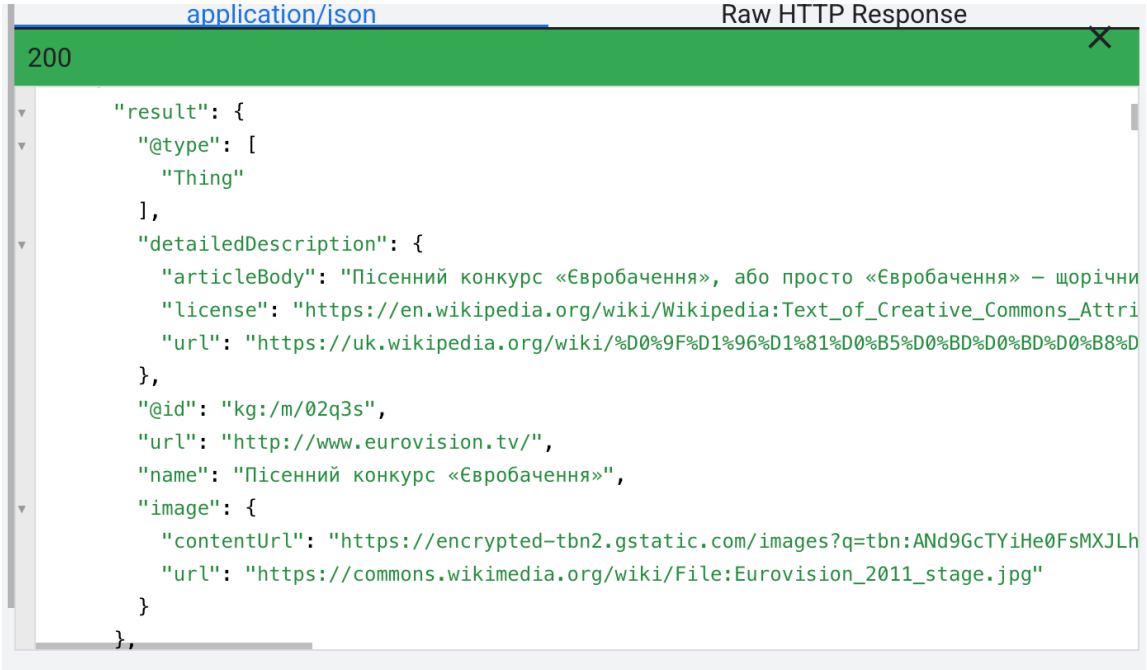

Рисунок 2.4 – Приклад запиту до бази знань від Google Knowedge Graph

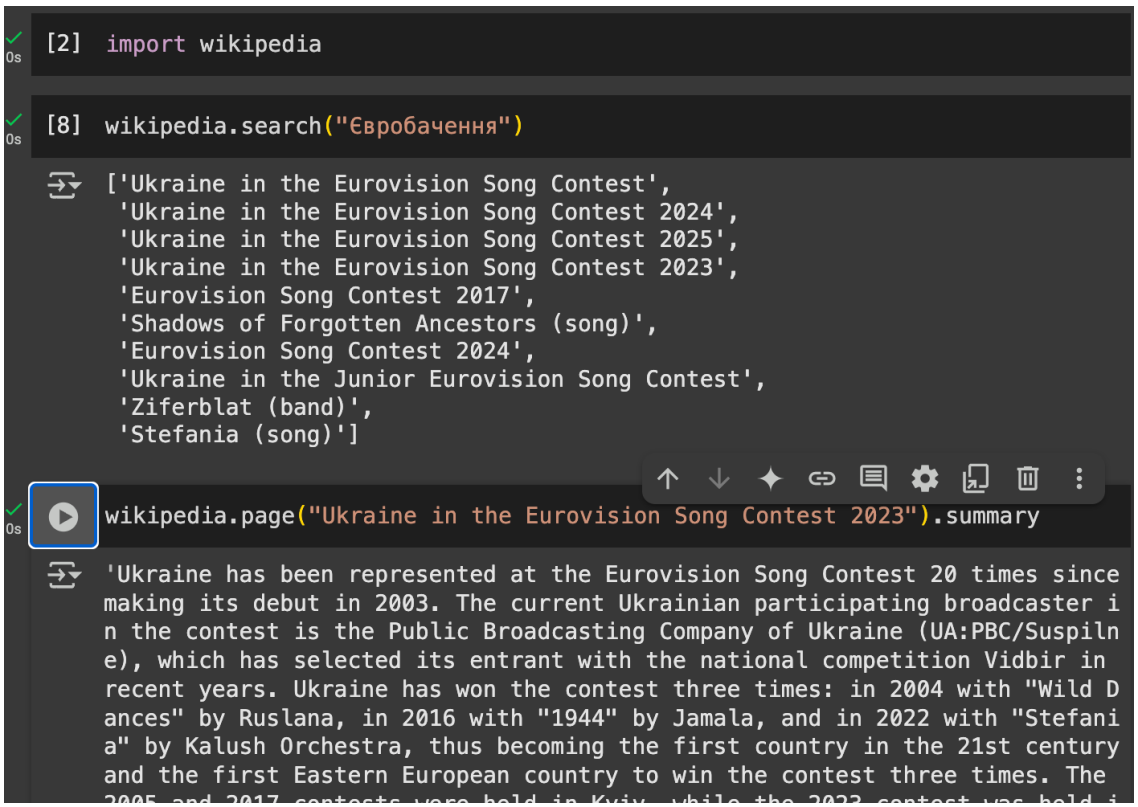

Рисунок 2.5 – Приклад запиту до всесвітньої бази знань Wikipedia

2.1.2. Структура даних для навчання

Серед існуючих наборів даних можна виділити наступні недоліки:

– відсутність достовірних перевірених даних англійською мовою після 2020 року;

– переважна кількість наборів пов’язаних з медициною обмежується даними про COVID-19;

– зразки української мовою майже відсутні;

– нестача чіткої системи класифікації для більше ніж двох класів (відмінність між багатьма класами часто надто розмита);

– неповнота ознак та/або перемішені «брудні» дані;

– відсутність даних у технічних, наукових сферах.

Дані для навчання та тренування моделі мають поєднувати в собі різні теми, контексти а тому існує потреба в об’єднанні декількох наборів даних.

Ключовими аспектами при виборі датасетів є: актуальність даних, доступність, чистота даних, варіативність. Проаналізувавши існуючі набори даних було прийнято рішення об’єднати сфери політики, медицини, загальних чуток та жартів і використати для навчання наступні датасети: News Headlines Dataset For Sarcasm Detection, Tweets with Sarcasm and Irony, FakeHealth, CoAID, FaceCovid Politifact Fact Check, CoAID.

Також в якості додаткових екземплярів українською мовою мають бути використані платформа FakeCheck та «Ґвара медіа». З кожного з них буде використана лише частина даних. Всі ці набори даних мають різні ознаки і систему позначок, а тому мають бути приведені до спільного формату, який міститиме наступні ознаки: заголовок, текст, автор, джерело, дата. Деякі з ознаки можуть бути відсутніми для деяких екземплярів, в такому випадку буде використано значення «невідомо» яке відповідатиме 0. На табличці 2.1 описано детальніше структура даних.

Таблиця 2.1 – Опис структури набору даних.

Назва	Значення	Опис
Заголовок	Текст або пустий рядок	Текст заголовка, може бути відсутнім лише за умови наявності.
Текст вмісту	Текст або пустий рядок	Власне інформації, якщо повний текст відсутній, то має бути заголовок.
Автор	Текст або пустий рядок	Ім’я або нікнейм автора, може бути відсутнім.
Джерело	Текст або пустий рядок	В якості джерела може бути конкретна соціальна мережа, сайт.
Дата	UTC формат	Дата публікації / заяви. Може бути відсутня.
Мова	0 – Англійська, 1 – Українська.	Мова тексту. Обов’язкове поле.

Таким чином багато ознак цієї структури можуть бути відсутніми. Перша причина цього – це відсутність деяких даних у вибраних навчальних даних, друга – потреба пристосування моделі до реальних даних, адже користувач, який хоче перевірити інформацію не завжди має додаткові ознаки про неї, іноді все що є – це текст.

В якості класів пропонується такі три позначки: «правда», «брехня» і «сатира». Кожен лейбл з оригінальних наборів даних має бути конвертованим.

2.2. Структура системи

В цьому дослідженні пропонується детально розглянути гібридну систему розпізнавання дезінформації в медіа контенті, а саме використання внутрішніх та зовнішніх даних. Аналіз зовнішнього контексту завжди показує кращі результати, завдяки дійсній перевірці фактів. Такий алгоритм не буде уразливим до простої підміни понять зі збереженням стилю, проте потребувати великих ресурсних витрат і більшої кількості достовірної інформації. До того ж на ранніх етапах створення новин чи постів може бути недостатньо існуючої контекстної інформації. Саме тому існує потреба в перевірці суто внутрішньої інформації. Для того, щоб ефективно поєднати ці два підходи, залишаючи їх двома окремими компонентами необхідно побудувати мультиагентну систему.

2.2.1 Мультиагентна система

Мультиагентна системи (МАС) – це структура, в яких кілька незалежних агентів, кожен з яких здатний самостійно приймати рішення, працюють разом для досягнення складних цілей. Ці агенти можуть співпрацювати, координувати або навіть конкурувати, залежно від цілей.

Основою системи є агенти – сутності, моделі ШІ, призначені для виконання певних завдань. Відповідно мультиагентні системи включають кілька таких моделей, які взаємодіють між собою, щоб вирішувати проблеми більш ефективно, ніж система з одним агентом. Перевагами таких систем є вирішення більш складних задач, масштабованість та динамічне адаптування. МАС вирішують таку низку проблем поодиноких агентів:

– складна багатопоточність;

– обмеження контексту;

– ускладнення прийняття рішень при використанні декількох моделей;

– дефіцит спеціалізації [44].

В рамках цього дослідження пропонується використати МАС, що складається з двох агентів: аналіз текстового вмісту та аналіз зовнішнього контексту. Дані користувача передаються на вхід обом агентам, але блок зовнішнього аналізу також на вхід отримує оцінку внутрішнього агенту. Результат роботи другого блоку повертається користувачу як кінцева відповідь. На рисунку 2.6 зображено схему вище описаної системи.

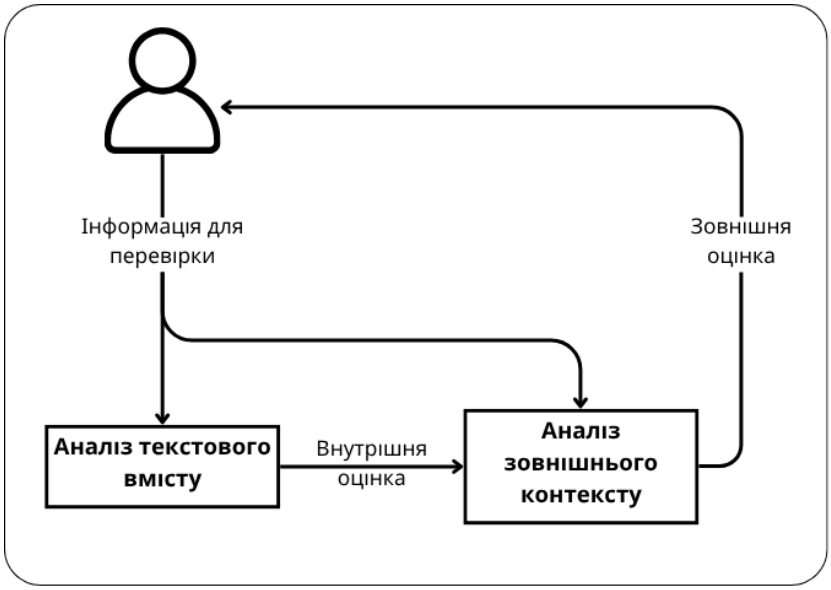

Рисунок 2.6 - Схематичне зображення запропонованої архітектури мультиагентної системи для розпізнавання дезінформації

2.2.2 Аналіз текстового вмісту

Агент, що аналізуватиме внутрішній зміст, на вхід отримує лише заголовок та основний текст. В якості ознак навчання пропонується розглянути емоційну оцінку та контекстний вектор. Для глибшого розуміння векторизація має бути контекстною та мати можливість використання української мови. Враховуючи ці критерії для вилучення ознак з тексту пропонується використати згадану вище та часто використовану в дослідженнях модель трансформер BERT. На платформі Hugging Face було знайдено ваги для цієї моделі попередньо натреновані на 104 найпопулярніших мовах на даних платформи Wikipedia в 2019 році, серед яких є і українська [45]. Таким чином це буде перенесне (трансферне навчання).

Основна ідея трансферного навчання полягає в тому, що ознаки, вивчені глибокою нейронною мережею на великому та різноманітному наборі даних, часто узагальнюються та корисні для інших пов'язаних завдань. Після навчання моделі на цих універсальних колосальних даних результуючі ваги використовуються в якості ініціалізуючої позиції для подальшого навчання для специфічної задачі. В нашому контексті трансферне навчання вирішує декілька проблем: маленький набір даних українською мовою, обмежені обчислювальні потужності та обмежений час. Додатковими перевагами є в середньому вища точність, а також вища стійкість до перетренування (англ. overfiiting).

Модель BERT (англ. Bidirectional Encoder Representations from Transformers – двонаправлені представлення кодувальників трансформера) – це яскравий приклад трансферного навчання в NLP. Це модель на основі трансформатора, попередньо навчена на величезному корпусі текстових даних (BooksCorpus та англійська Вікіпедія) з використанням двох завдань навчання без вчителя:

– модель маскованої мови (англ. MLM), представляє навчання передбачати випадкові замасковані слова в реченні. Це змушує модель вивчати багаті контекстуальні представлення слів на основі їхнього навколишнього контексту;

– прогнозування наступного речення (англ. NSP), задача навчитися передбачати, чи є два речення послідовними в оригінальному тексті. Це допомагає моделі розуміти зв'язки між реченнями.

BERT є моделлю яка містить лише кодувальну частину трансформера. Блок кодувальника в трансформері відповідає за прийом вхідної послідовності та створення багатих числових векторних представлень для кожного токена (текстової одиниці). Моделі, що містять тільки кодувальник (рисунок 2.7), опускають декодувальник і об'єднують кілька кодувальних частин, щоб створити єдину модель. Літера N позначає кількість кодувальників, для обраної багатомовнох моделі BERT це значення дорівнює 12.

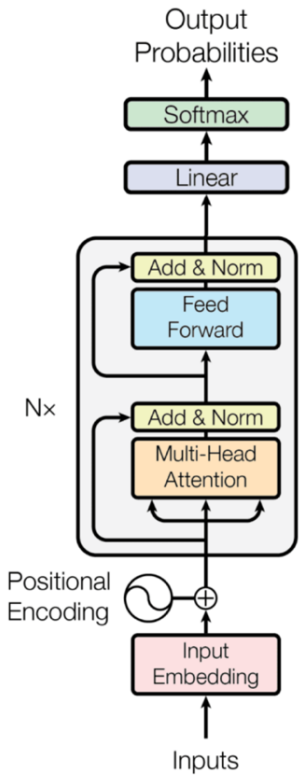

Рисунок 2.7 – Загальна архітектура мережі лише з кодувальник [11]

Ці моделі не приймають підказки як такі, тільки вхідну послідовність для прогнозування (наприклад, прогнозування відсутнього слова в послідовності). Багаті векторні представлення, створені блоками кодера, дають BERT глибоке розуміння вхідного тексту.

Літера N позначає кількість кодувальників, для обраної багатомовнох моделі BERT це значення дорівнює 12.

Однією з головних переваг моделі, що працює на кодувальника, що відрізняє її від декодувальних мереж – це двонаправленість. Двонаправленість означає, що кожне слово у вхідній послідовності може отримувати контекст як від попередніх, так і від наступних слів, тобто механізм уваги може звертати увагу на попередні та наступні лексеми для кожного слова. Це допомагає BERT робити більш обґрунтовані прогнози. Найважливішим елементом трансформера є механізм самоуваги, що дозволяє фіксувати довгострокові залежності без послідовної обробки. Так названа «багатоголова увага» в трансформерах підвищує продуктивність моделі, дозволяючи їй одночасно зосередитися на різних аспектах вхідних даних. Алгоритмічно механізм уваги представляє собою елемент, який на вхід приймає матрицю запитів Q (поточна інформація, що оброблюється) розмірністю (N, матриця ключів K (фактори, що можуть бути релевантними для Q), матриця значень V (значення факторів), далі відбувається ряд математичних перетворень, в основі яких лежать скалярному добутку векторів та функції softmax:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \text{ [11]}, \quad (2.1)$$

де d_k – розмірність одного зразка K і Q;

Завдяки цьому елементу модель здатна взважено обраховувати вплив різного контексту на поточну інформацію. На рис. 2.8 схематично зображено механізм масштабованої уваги. «Багатоголова увага»

представляє собою поєднання таких елементів уваги, схема такої системи зображена на рисунку 2.9.

Процес вбудовування вхідних даних для BERT складається з трьох етапів: позиційне вбудовування, вбудовування сегментів і вбудовування токенів. Позиційна інформація входить у вбудовування для кожного токена і для BERT є фіксованою. Це означає, що BERT обмежений 512 токенами у вхідній послідовності як для базової, так і для великої моделі.

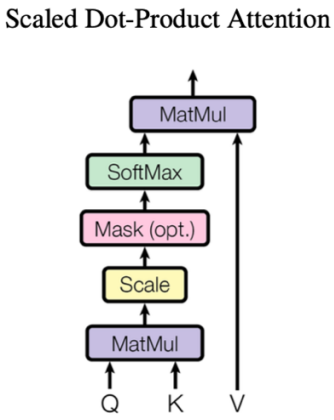

Рисунок 2.8 – Схематичне зображення механізму масштабованої уваги

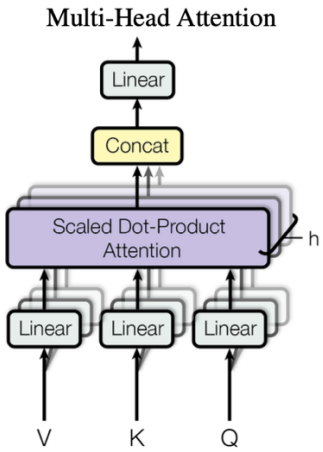

Рисунок 2.9 – Схематичне представлення багатоголової уваги

При вбудовуванні також додаються вектори, що кодують сегмент, до якого належить кожен токен. Всі токени у вхідних даних, де є лише один роздільний токен [SEP] вважаються такими, що належать до сегмента A. Для завдань, де таких токенів більше (для задач передбачення наступного тексту) – всі токени після другого [SEP] позначаються як сегмент B. Власне вбудовування кожного токена потім додається до його позиційних і сегментних векторів, щоб створити остаточне вбудовування, яке буде передано механізмам самоуваги в BERT для додавання контекстної інформації.

Попередньо навчений BERT можна використати для рішення таких задач:

– класифікація емоційного забарвлення, та класифікація тексту: шар класифікації додається після об'єднаного виводу (представляє все речення) та дотреновується на нових даних;

– відповіді на запитання;

– розпізнавання іменованих сутностей: класифікаційний шар додається на рівні токенів для прогнозування типу сутності (особа, організація, місцезнаходження) для кожного токена в реченні.

Попередня обробка інформації для BERT не тільки має привести дані до токенів правильної довжини, але також додає специфічні особливі токени. Загалом BERT має п'ять спеціальних токенів:

– [PAD] (ідентифікатор = 0) – токен-заповнювач, який використовується для доведення загальної кількості токенів у вхідній послідовності до 512;

– [UNK] (ідентифікатор = 100) – невідомий токен, який використовується для позначення токена, що не входить до словника BERT;

– [CLS] (ідентифікатор = 101) – токен класифікації, який повинен бути на початку кожної послідовності, незалежно від того, чи використовується він, чи ні. Цей токен узагальнює інформацію про клас для

завдань класифікації і може розглядатися як агреговане представлення послідовності;

– [SEP] (ідентифікатор = 102) – токен-роздільник, який використовується для розрізнення двох сегментів в одній послідовності вхідних даних (у прогнозуванні наступного речення). Очікується, що в кожній послідовності введення буде принаймні один токен [SEP], максимум два;

– [MASK] (ідентифікатор = 103) – токен маски, який використовується для навчання BERT на завданні моделювання маскованій мови або для виконання виведення на маскованій послідовності.

Дотренування моделі передбачає навчання лише лінійного шару: невеликої нейронної мережі з прямим поширенням, яку часто називають класифікаційною головою або просто головою. Ваги та зміщення в решті моделі, тобто в основній частині залишаються незмінними або замороженими. Для перетворення текстового вбудовування в один з трьох класів необхідно додати шар-класифікатор, це може бути або лінійний шар, що є стандартною опцією, або інший класифікатор. Пропонується розглянути три варіанти, такі як лінійний шар.

Лінійний шар є «повністю зв'язаним шаром» або «щільним шаром», який незважаючи на свою простоту, є дуже ефективним, оскільки дозволяє нейронній мережі вивчати складні взаємозв'язки між входами та виходами в поєднанні з нелінійними функціями активації. Лінійний шар для класифікації на вхід приймає N параметрів і має повернути M значень, що відповідає кількості класів. Для цього він виконує множення матриці з вагами на вектор ознак та додає вектор зміщення (формула 2.2). Активація цих вихідних даних поверне ймовірності для кожного класу.

$$y = W \cdot x + b \text{ [11]}, \tag{2.2}$$

де y – це вихідний вектор довжиною M;

W – матриця (N, M) з вагами, що навчаються;

x – вхідний вектор довжиною N;

B – вектор зсуву.

2.2.3 Аналіз зовнішнього контексту

Повна перевірка фактів потребує додаткових знань з якої вона братиме контекст. Ці знання можуть бути частиною моделі, якщо мова йде про LLM, проте це потребує колосальних витрат ресурсів при навчанні, а також дуже великого об’єму пам’яті. Таким чином ці знання мають бути винесені в окрему частину, і кожен раз при виклику моделі, вона має звертатися до цієї бази знань з пошуковим запитом, який допомагатиме визначити релевантний контекст. Це може бути реалізовано декількома способами. Перший – кожен раз робити пошук серед надійних джерел або просто в інтернеті, обробляти цю інформацію, й забувати. Це підхід не потребує сховища, та простіший в реалізації. Єдина його проблема це необхідність отримувати та обробляти часто одну й ту саму інформацію. Другий підхід має на увазі використання додаткового сховища, яке зберігає оброблені дані та оновлюється по мірі створення нових запитів в систему в разі нестачі існуючого контексту. Такий спосіб працюватиме ефективніше при постійних запитах, оскільки зберігатиме дані, проте вимагає великого обсягу пам’яті та додаткових алгоритмічних рішень. Спільним для обох способів є пошук нової інформації.

Основним джерелом в цьому контексті можуть бути платформа Wikipedia, та Google Knowledge Graph, ці дані мають певний кредит довіри. З іншого боку повний вихід в інтернет здатний принести більше контексту і прикладів неправдивих новин, тому його теж варто розглянути.

Коли модель отримує на вхід інформацію, вона має прийняти рішення який запит для пошуку контексту їй треба зробити. Це доволі складний та складений процес. Оскільки текст може бути досить великим, не достатньо

просто вписати його в пошуковий запит, треба узагальнити його зміст. Спочатку дані очищаються від зайвої інформації, для спрощення подальшої обробки, далі необхідно створити векторну версію тексту чи його частини і виділити основні сутності, які далі використовуватимуться для побудови запиту. Також можна визначити події чи стосунки, проте це є не обов'язковим. Векторизоване представлення тексту може бути використано для звернення до власної бази знань, якщо така існує, для швидшого пошуку. Для пошуку в інтернеті ця опція не підійде, проте за наявності повного трансформеру вектор може бути перехідним вбудованим представленням, що передається до декодувальника, натренованого генерувати запити, чи просто скорочення. Ключові слова потрібні для пошуку у мережі. Комбінуючи сутності між собою можна отримати релевантну інформацію. Наприклад ми маємо новину: «Британський журнал «The Guardian» висміяв у новому випуску зачіску президента Франції», ключовими словами можуть бути «The Guardian» і президент Франції, відповідно пошук будуватиметься на поєднанні цих сутностей. Для більш широкого контексту можна зробити три окремі запити: по кожній сутності та за поєднанням. Схематичне зображення цього процесу зображене на рисунку 2.10.

Задача вилучення ключових слів може бути вирішена статистично (підрахунок частоти вживання слів), алгоритмічно (на основі правил та патернів) а також за допомогою глибинного навчання. Прикладом статистичного алгоритму є RAKE (англ. – rapid automatic keyword extraction – повторювальне вилучення ключових слів). Цей алгоритм потребує спочатку видалення стоп слів, і розділення тексту на токени, часто для цього використовується ділення за видаленими стоп словами. Далі для кожного елемента обраховується частота та градус, де перше це відповідно частота повторювальності в тексті, а друге – кількість слів, у зв'язці з якими токен з'являється.

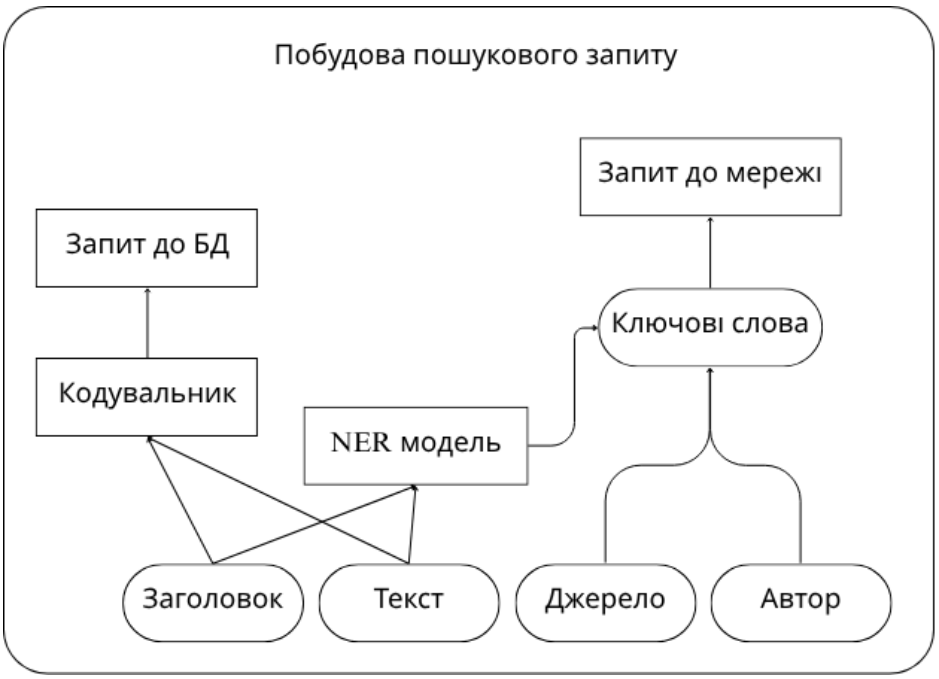

Рисунок 2.10 – Схематичне зображення побудови пошукового запиту

Алгоритми на патернах використовують спеціальні словники для розпізнавання частин мови, для пошуку патернів типу «прикметник іменник», або вилучають власні назви, такі як люди, цей процес називається NER (англ.– Named-Entity-recognition). Також існує графовий підхід, де кожен токен представляється, як вершина, а співіснування в тексті між токенами ребрами. Найбільш ефективним для вилучення ключових фраз є глибинні нейронні мережі. Наприклад спеціальна версія KBERT працює на основі архітектури звичайного BERT, проте спочатку обраховується вбудовування всього документу, далі обираються слова-кандидати з використанням n-грам, або патернів на основі частин мови, для них також обраховуються вбудовування, далі для кожного значення вбудовування фрази обраховується семантична подібність за допомогою косинуса подібності та обираються найкращі. Ця математична дія дозволяє порівняти семантичну схожість. Формула косинусу подібності виглядає наступним чином:

$$\cos(\theta) = \frac{A \cdot B}{\|A\| * \|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}, \quad (2.3)$$

де A і B – це вектори, що порівнюються.

Після будування запиту, та отримання результатів пошуку, необхідно отримати найбільш релевантну інформацію. Для цього вже отриманий текст можна теж векторизувати та порівняти за допомогою косинусу подібності. Для перевірки правдоподібності інформації, кожна нова стаття має бути передана модулю лінгвістичного аналізу для оцінки на дезінформацію (кожній такі статті надаватиметься ймовірність замість чіткого позначення класу).

Для кращого семантичного розуміння для перевірки фактів таку велику кількість даних краще представляти у вигляді графової структури, тобто ГЗ. При цьому ГЗ має бути побудованим автоматично за допомогою ШІ. Цей процес є високорівневою послідовністю, що складається з таких компонентів: NER, вилучення зв’язків між сутностями (англ. RE – relation Extraction), вилучення подій і розмиття даних.

Після попередньої обробки текстової інформації необхідно виділи об’єкти з власними назвами, тобто персони, місця та організації, для цього необхідна певний набір даних з чіткою класифікацією таких сутностей і натренована на ньому модель, найкраще підходять для цього моделі на базі BERT.

Наступним кроком необхідно вирішити питання кореферентності, тобто пов’язати займенники з виділеними власними назвами, а також різні назви одного об’єкту. Наприклад у тексті «Джері Хейл випустила нову пісню, вона називається «Сніг», виконавиця розповіла про..» слово «виконавиця» має посилатися на власну назву «Джері Хейл», в той час як займенник «вона» до людини в цьому контексті не стосується. Така задача Найкраще вирішується за допомогою трансформерних моделей типу SpanBERT чи Longformer а також моделі e2e (англ. end-to-end – від початку

до кінця), особливість останньої в тому, щоб не розділяти процес на етапи, а вирішувати задачу цілісно.

Далі слідує визначення зв’язків, тобто структури суб’єкт-відносини-об’єкт (триплет), для цього також найкраще підходять трансформерні моделі, які на вхід приймають перелік сутностей, та повертають взаємини. Зручним є підхід одночасного виявлення сутностей та їх зв’язків, тут також підійде SpanBERT. Прикладом виходу з BERT моделі може бути: [CLS] Стів Джобс [E1] заснував [E2] «Apple» [/E2] в 1976. [/E1]. Кожен результат знайдених відносин має бути представлений у вигляді триплетів.

Для побудови самого графа можна використати наступні правила:

– кожна власна сутність перетворюється на вершину, з атрибутом типу, дати створення (оновлення), векторним вбудовуванням, за наявності відносин типу «народився», «був заснований», «помер» і «закрився» додавати дані про дату;

– короткий опис подій, твердження зберігати як вершини, з атрибутом вмісту, векторним представленням, датою твердження, датою створення чи оновлення, оцінкою правдоподібності та зв’язками зі згаданими сутностями, та ребрами з автором і джерелом.

Окремо варто реалізувати систему «пеналті»: в кожної сутності має бути два лічильники, що відповідають за частоту згадування у фейковій новині, та частоту авторства неправдивої інформації, інакше кажучи показник жертви дезінформації та джерела. Другий показник має застосовуватися лише до персон та організацій. Ці значення мають оновлюватися за рахунок зворотнього зв’язку, після класифікації оригінальної новини, модель має знову звернутися до ГЗ та додати нову інформацію і оновити показники пов’язаних з нею сутностей.

При отриманні запиту на контекстну інформацію має бути три умови для виходу в мережу з метою отримання нової інформації:

– недостатньо релевантна інформація (косинусна відстань більше порогового значення);

– не актуальна фнформація (дата оновлення менша за порогове значення);

– знайдено не всі запитані сутності.

Перевірка фактів має починатися з конвертування вхідної інформації у формат ГЗ, тобто обробити вхідну новину, як таку, що була знайдена в мережі. Для роботи з графом для порівняння фактів потрібна GNN, враховуючи гетерогенність даних в ГЗ та масштаби розмірів, до яких граф може вирости, хорошим архітектурним рішенням є модель HGT (англ. Heterogeneous Graph Transformer – гетерогенний графовий трансформер), проте навчання такого рішення вимагає надзвичайно великих ресурсних витрат, тож наступним вибором є HAT (англ. Heterogeneous graph Attention network – гетерогенна графова мережа з механізмом уваги) [47]. Архітектуру було представлено в 2019 році, вона ґрунтується на моделі GAT, проте враховує мета-шляхи між вершинами, тобто працює для неоднорідних даних, де вершини та ребра мають свій тип. Мета-шлях це шлях типу $A_1 \overset{R_1}{\rightarrow} A_2 \overset{R_2}{\rightarrow} A_1$, тобто де ребра мають тип. HAT використовує увагу на рівні вузла та семантичну. Увага на рівні вузла може вивчити важливість сусідів на основі мета-шляхів для кожного вузла в гетерогенному графі та агрегувати представлення цих значущих сусідів для формування вбудованого вузла. Для кожного типу вершини ϕ_i має бути представлена спеціальна матриця перетворення $M_{\phi i}$ для представлення різних типів вершин до спільного простору ознак. Таким чином формула для обрахування важливості мета-шляху ϕ для пари вершин i та j обраховується наступним чином [47]:

$$e_{ij}^{\phi} = att_{node}(h_i^{`}, h_j^{`}; \phi), \tag{2.4}$$

$$a_{ij}^{\phi} = softmax(e_{ij}^{\phi}) \tag{2.5}$$

де $h_i^{`}$ та $h_j^{`}$ є уніфікованими векторними представленнями вершин і та j відповідно, та обраховується як $h_i^{`} = M_{\phi i} \cdot h_i$,

a_{ij}^{ϕ} – нормалізований вектор вагових коєфіцієнтів.

Увага на рівні вузла повторюється K разів та об'єднує вивчені вбудовування як семантично-специфічне вбудовування:

$$z_i^{\phi} = \Big\|_{k=1}^{K} \sigma\left(\sum_{j \in N_i^{\phi}} a_{ij}^{\phi} \cdot h_i^{`}\right), \tag{2.6}$$

де z_i^{ϕ} – семантично-специфічне вбудовування,

N_i^{ϕ} – сусіди вершини і по мета-шляху ϕ.

Семантична увага призначена для вивчення важливості кожного мета-шляху. Скажімо існує група з P мета-шляхів тоді ваги цих шляхів можна обчислити наступним чином:

$$\left(\beta_{\phi_1}, \ldots, \beta_{\phi_p}\right) = att_{sem}\left(Z_{\phi_1}, \ldots, Z_{\phi_p}\right), \tag{2.7}$$

$$w_{\phi_p} = \frac{1}{|V|} \sum_{i \in V} q^T \cdot \tanh\left(W \cdot z_i^{\phi_p}\right) + b, \tag{2.8}$$

$$\beta_{\phi_p} = \frac{\exp\left(w_{\phi_p}\right)}{\sum_{p=1}^{P} \exp\left(w_{\phi_p}\right)}, \tag{2.9}$$

де β_{ϕ_p} – ваги для мета-шляху ϕ_p;

w_{ϕ_p} – важливість мета-шляху ϕ_p;

|V| – кількість вершин;

q – вектор уваги семантичного рівня;

W – матриця вагів;

b – вектор зсуву.

Таким чином обрахування результуючого вбудовування Z виглядає так:

$$Z = \sum_{p=1}^{P} \beta_{\phi_p} \cdot Z_{\phi_p}. \quad (2.10)$$

Тобто HAT працює наступним чином: усі типи вузлів проектуються в єдиний простір ознак, а вагу пари вузлів на основі мета-шляхів можна дізнатися за допомогою уваги на рівні вузлів, далі йде процес спільного навчання ваг кожного мета-шляху та об'єднання семантично-специфічного вбудовування вузлів за допомогою уваги на семантичному рівні. В кінці обчислення втрат та наскрізної оптимізації. На рисунку 2.11 зображено архітектуру HAT представлену в 2019 році.

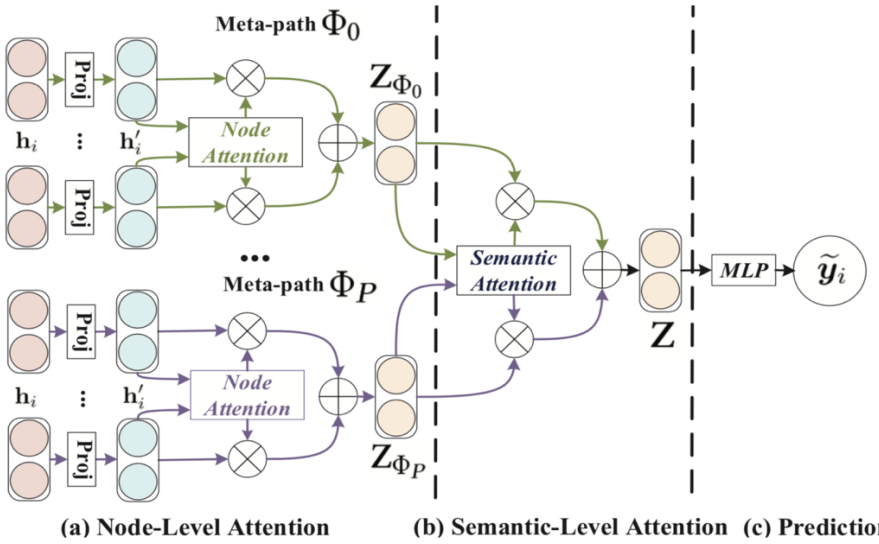

Рисунок 2.11 – Схематичне представлення будови HAT 2019 р. [47]

HAT працюватиме менш ефективно на надто великих графах, тому на вхід необхідно подавати лише дійсно релевантну інформацію. Можливим рішенням є використати виділені на вхідних даних сутності для пошуку

подібних сутностей у ГЗ, й виокремити для класифікації лише вершини та ребра, що пов’язані з цими вершинами.

2.3 Висновки

Таким чином, проєктування бази знань для задачі розпізнавання дезінформації висвітлило найбільші проблеми серед існуючих наборів даних. А саме нестачу більш нових та варіативних екземплярів та майже відсутність наборів даних українською мовою. Для вирішення цієї проблеми можна використати розширення існуючих даних шляхом веб-парсингу популярних надійних ресурсів, присвячених перевірці інформації таких як PolitiFact, FactCheck та медіа «Ґвара». В якості навчальних даних необхідно використати комбінацію наборів різних сфер та класифікацій. Пропонується також система класифікації в три лейбли: правда, брехня, сарказм.

Рішення має представляти собою мультиагентну систему, що складається з двох основних частин: аналізу внутрішнього контексту, тобто лінгвістична оцінка та зовнішнього контексту, тобто фактична перевірка інформації. Схему запропонованого рішення представлено на рисунку 2.12. В якості першого модуля пропонується використати модель, що базується на використанні трансформерного кодувальника BERT з вагами натренованими на великій кількості мов, серед яких є українська. Попередньо треновану модель необхідно використати для дотренування класифікаційного шару, що базується на лінійному шарі нейронної мережі.

Модуль перевірки фактів потребує побудови бази даних на основі графу знань, яка має оновлюватися за умови нестачі актуального та релевантного контексту, в якості джерела інформації може використовуватися Вікіпедія, Google Knowledge Graph або вихід в інтернет за допомогою пошукового двигуну.

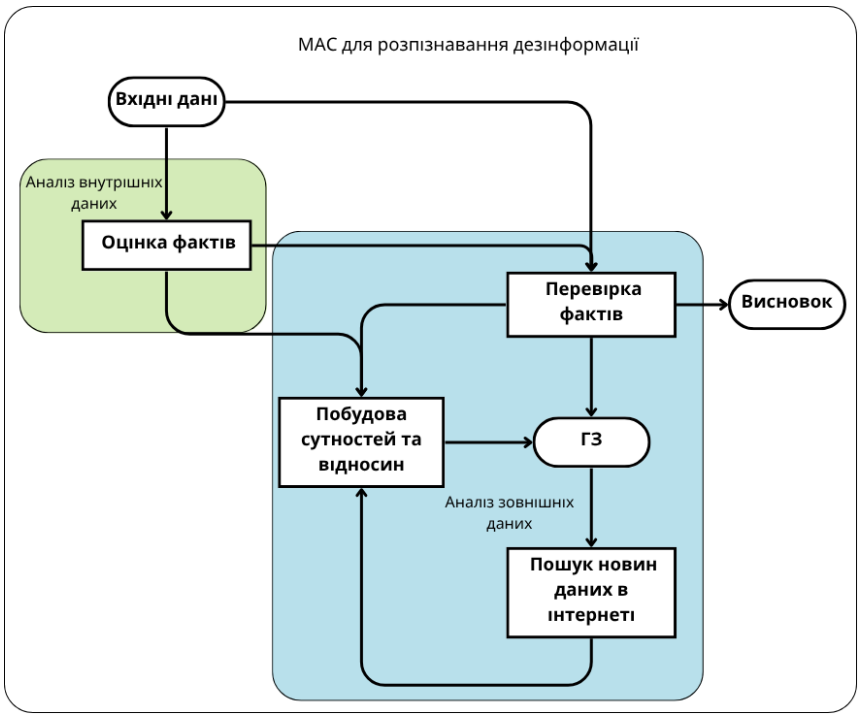

Рисунок 2.12 – Схематичне зображення запропонованої системи для вирішення задачі розпізнавання дезінформації в медіа-контенті

Модуль перевірки внутрішньої інформації має використовуватися не тільки для попередньої оцінки вхідних даних, але також для оцінки нових знайдених статей. Модель перевірки фактів має працювати на основі архітектури HAT.

3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ

3.1 Розробка агенту з аналізу внутрішнього контенту

В якості експерименту пропонується побудувати дві моделі для аналізу текстової інформації на наявність маніпуляцій: багатомовний попередньо тренований BERT та класифікатор на базі випадкових дерев з використанням в якості характеристик вектори TF-IDF, показник читабельності та емоційне забарвлення тексту.

В якості навчальних даних пропонується використати поєднання набору даних з текстами від PolitiFact, комедійні тексти від The Onion, перевірки фактів від Ґарда медіа і платформи FactCheck від Google. Для більшого зрівняння англійської та української мови у даних, частину англійських текстів було перекладено українською за допомогою штучного інтелекту. В якості класів використовуються лейбли «дезінформація», «нейтрально», «сатира».

Модель на базі набору різних ознак має використовувати семантичну оцінку. Для цього на платформі Hugging face було знайдено натреновану мережу на основі BERT, яка розуміє українську та має 5 різних класів емоцій: дуже негативна, негативна, нейтральна, позитивна, дуже позитивна [48].

Для перетворення текстів на індекси TF-IDF необхідно попередньо прибрати стоп слова, для англійської мови є функції, що роблять це автоматично в багатьох бібліотеках Python, для української було знайдено такий список в інтернеті, він містить 176 слова. Лематизація була досягнута за рахунок видалення популярних префіксів, суфіксів та закінчень. Це потенційно може обрізати слово неправильно, проте його основа з великою ймовірністю все одно лишиться унікальною.

Третьою ознакою є читабельність (англ. RI – readability index). Вона визначається через кількість складних слів. Існує декілька формул за якими

обраховується цей показник, в цьому дослідженні для було використано дві різних формул, одна для англійської (3.1) й інша для української (3.2):

$$RI_{\text{англ}} = 0.4 * (\frac{\text{кількість слів}}{\text{кількість речень}} + 100 * \frac{\text{кількість складних слів}}{\text{кількість слів}}), \quad (3.1)$$

$$RI_{\text{укр}} = 0.5 * \left(\frac{\text{кількість слів}}{\text{кількість речень}}\right) + 0.4 * (100 * \frac{\text{кількість складних слів}}{\text{кількість слів}}), \quad (3.2)$$

де кількість складних слів позначає слова з більше ніж 3 голосними.

В якості класифікатора використовується модель випадкового лісу, що є ансамблем дерев, що приймають рішення. Реалізація моделі була взята з Python бібліотеки Scikit-lear.

Модель на основі BERT використовує попередньо натреновану версію, взяту з платформи Hugging face, що включає в себе розуміння української мови. Результуюче вбудовування має довжину в 768 символів, розмір одного вхідного токена не може бути більшим за 512. Також ця версія моделі не є чутливою до великих літер, оскільки дані, що використовуються є неоднорідними, та часто їхнім джерелом є інтернет мережі, де люди не завжди використовують великі літери. Існуюча проблема з обмеженням у 512 токенів для базової моделі не є актуальними для даної навчальної вибірки, у разі занадто великого тексту, він обрізається до останнього речення в межах норми. На рисунку 2.9 зображено структуру моделі BERT.

Класифікатор, що додається зверху основної архітектури складається з лінійного шару, що має вхід у 768 параметрів, та вихід у 3, що відповідає трьом різним класам. В якості функції активації використовується функція softmax. В якості функції втрат використовується крос ентропія. Для реалізації моделі було використано Python бібліотеки Pytorch та Transformers. Результати тренування обох моделей зображені на рисунку 3.2.

```
from transformers import BertTokenizer, BertModel
from torchsummary import summary

tokenizer = BertTokenizer.from_pretrained('bert-base-multilingual-cased')
model = BertModel.from_pretrained("bert-base-multilingual-cased")

print(model)
```

```
BertModel(
  (embeddings): BertEmbeddings(
    (word_embeddings): Embedding(119547, 768, padding_idx=0)
    (position_embeddings): Embedding(512, 768)
    (token_type_embeddings): Embedding(2, 768)
    (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
    (dropout): Dropout(p=0.1, inplace=False)
  )
  (encoder): BertEncoder(
    (layer): ModuleList(
      (0-11): 12 x BertLayer(
        (attention): BertAttention(
          (self): BertSdpaSelfAttention(
            (query): Linear(in_features=768, out_features=768, bias=True)
            (key): Linear(in_features=768, out_features=768, bias=True)
            (value): Linear(in_features=768, out_features=768, bias=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
          (output): BertSelfOutput(
            (dense): Linear(in_features=768, out_features=768, bias=True)
            (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
        )
        (intermediate): BertIntermediate(
          (dense): Linear(in_features=768, out_features=3072, bias=True)
          (intermediate_act_fn): GELUActivation()
        )
        (output): BertOutput(
          (dense): Linear(in_features=3072, out_features=768, bias=True)
          (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
          (dropout): Dropout(p=0.1, inplace=False)
        )
      )
    )
  )
  (pooler): BertPooler(
    (dense): Linear(in_features=768, out_features=768, bias=True)
    (activation): Tanh()
  )
)
```

Рисунок 3.1 – Структура багатомовної моделі BERT від Hugging Face

```
print(f"F1 score for BERT:{bert_f1_score},\nF1 score for classic algorithm: {classic_f1_score}")
```

```
[124] ✓ 0.0s
```

```
F1 score for BERT:80.42,
F1 score for classic algorithm: 70.31
```

Рисунок 3.2 – Результат навчання моделі обох моделей

З результатів тренування видно, що рішення на базі трансформерного кодувальника має вищі показники міри F1. На рисунку 3.3 зображено приклад класифікації новини, згенерованої за допомогою штучного інтелекту.

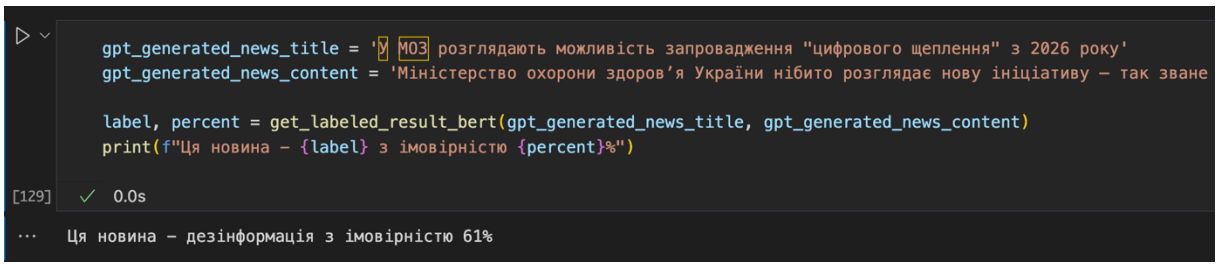

Рисунок 3.3 – Результат класифікації для новини згенерованої штучним інтелектом

Новину було згенеровано за допомогою чату GPT з промтом «згенеруй правдоподібну фейкову новину». Модель на основі BERT впоралася з класифікацією. Проте враховуючи відсутність справжньої перевірки фактів, модель погано працює на коротких заголовках по типу «Синоптики обіцяють температуру понад 50 градусів завтра у Харкові», а також вразлива до підміни фактів в рамках правдивої новини.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи було проведено глибокий аналіз інструментів інтелектуального аналізу тексту в рамках задачі розпізнавання дезінформації в медіа-контенті. Після детального ознайомлення з найбільш релевантними дослідженнями в цій галузі було виділено такі слабкі місця, як ігнорування комедійних текстів в рамках класифікації брехні, відсутність експериментів в українському інформаційному сегменті, та потреба в гібридному підході. Було також визначено три основних принципи, на яких ґрунтуються рішення задачі розпізнавання брехні в інтернеті. Аналіз внутрішнього контенту бере до уваги виключно ознаки всередині вхідних даних, які представляють з себе текст, заголовок та іноді деякі метадані. Простіший в реалізації, швидкий, не потребує додаткової інформації проте не перевіряє в дійсності факт, а тому такі рішення є вразливими до адверсаріальних атак. Аналіз зовнішнього контексту передбачає вивчання релевантного контексту, взятого з додаткового джерела інформації. Такий підхід має набагато складнішу структуру, потребує більших даних та ресурсних витрат, проте здатен дійсно перевірити інформацію. Аналіз мережевих даних ґрунтується на вивченні користувацької взаємодії, він вивчає поведінку людей в інтернеті, їхні коментарі, та структури поширення даних. Такі алгоритми потребують багато інформації, яку важко дістати через приватність, не перевіряють факти на дійсність, проте здатні виявляти неправдиві пости на початкових етапах, і добре підходять для неконтрольованого навчання.

Серед існуючих навчальних даних значними проблемами є відсутність нових, актуальних даних, інформації, що репрезентувала би такі сфери, як навчання, наука, побут тощо, також екземплярів українською мовою. Для вирішення цієї проблеми пропонується використати дані з популярних надійних ресурсів, присвячених перевірці фактів, таких як

американській PolitiFact, міжнародний Google FactCheck та українське медіа «Ґвара» а також генерація новин за допомогою великих мовних моделей.

В рамках цього дослідження основною увагою стали алгоритми внутрішнього та зовнішнього аналізу інформації. Запропоновано використовувати мультиагентну систему, що містить модуль оцінки та модуль перевірки фактів. В якості можливих класів пропонуються позначки «брехня», «правда» та «сатира». Модуль оцінки фактів використовує для аналізу текст інформації та за допомогою трансформерної моделі лише з кодувальником BERT дає попередню оцінку щодо правдивості інформації. Модуль перевірки фактів має внутрішню базу даних у вигляді гетерогенного направленого графу знань, який використовується для додаткового контексту та заповнюється по мірі надходження нових запитів. Джерелом інформації може виступати Wikipedia, Google Knowledge Graph а також простий веб-пошук. Граф будується автоматично та оновлюється за відсутності релевантного актуального контексту. Вхідна інформація так само як і нова знайдена інформація отримує попереднє значення правдоподібності від агента оцінки фактів. Архітектурою агента перевірки фактів обрано модель HAT, що використовує алгоритм уваги на гетерогенному графі для отримання його вбудовування.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Journalism R. I. f. t. S. o. Digital News Report 2024. URL: https://reutersinstitute.politics.ox.ac.uk/digital-news-report/2024 (дата звернення: 05.03.2025).

2. Савченко О. В. Мас-медіа. *Енциклопедія Сучасної України*. URL: https://esu.com.ua/article-64254 (дата звернення: 05.03.2025).

3. Капелюшний В. П. Дезінформація. *Енциклопедія Сучасної України*. URL: https://esu.com.ua/article-21273 (дата звернення: 05.03.2025).

4. World Economic Forum. Global Risks Report 2024. URL: https://www.weforum.org/publications/global-risks-report-2024 (дата звернення: 06.03.2025).

5. Сім категорій дезінформації в Інтернеті | Репортери без кордонів Ресурс безпеки журналістів. *Репортери без кордонів Ресурс безпеки журналістів.* URL: https://safety.rsf.org/ukr/the-seven-categories-of-online-disinformation/ (дата звернення: 07.03.2025).

6. Wang W. Y. "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (Volume 2: Short Papers), м. Vancouver, Canada. Stroudsburg, PA, USA, 2017. URL: https://doi.org/10.18653/v1/p17-2067 (дата звернення: 10.03.2025).

7. Automatic Detection of Fake News / Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, Rada Mihalcea, 2017 р. URL: https://doi.org/10.48550/arXiv.1708.07104 (дата звернення 10.03.2025).

8. Omarzai F. Word2Vec (CBOW, skip-gram) in depth. *Medium*. URL: https://medium.com/@fraidoonomarzai99/word2vec-cbow-skip-gram-in-depth-88d9cc340a50 (дата звернення: 01.04.2025).

9. Yadav A. Word2Vec vs GloVe: Which Word Embedding Model is Right for You?. *Medium*. URL: https://medium.com/biased-algorithms/word2vec-vs-

glove-which-word-embedding-model-is-right-for-you-4dfc161c3f0c (дата звернення: 20.04.2025).

10. Kal. Word Embedding Using FastText. *Medium*. URL: https://medium.com/@93Kryptonian/word-embedding-using-fasttext-62beb0209db9 (дата звернення: 20.04.2025).

11. Thu P. P., Aung T. N. Implementation of Emotional Features on Satire Detection. *International Journal of Networked and Distributed Computing*. 2018. Т. 6, № 2. С. 78. URL: https://doi.org/10.2991/ijndc.2018.6.2.3 (дата звернення: 10.03.2025).

12. Attention Is All You Need / A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin *31st Conference on Neural Information Processing Systems*, Long Beach, CA, USA 2017.

13. Thu P. P., Aung T. N. Implementation of Emotional Features on Satire Detection. *International Journal of Networked and Distributed Computing*. 2018. Т. 6, № 2. С. 78. URL: https://doi.org/10.2991/ijndc.2018.6.2.3 (дата звернення: 26.04.2025).

14. Fake news detection via NLP is vulnerable to adversarial attacks / Z. Zhou та ін. *11th international conference on agents and artificial intelligence*, м. Prague, Czech Republic, 19–21 лют. 2019. URL: https://doi.org/10.5220/0007566307940800 (дата звернення: 10.03.2025).

15. Tida V. S., Hsu S., Hei X. A unified training process for fake news detection based on finetuned bidirectional encoder representation from transformers model. *Big data*. 2023. URL: https://doi.org/10.1089/big.2022.0050 (дата звернення: 01.04.2025).

16. Context-Driven satire detection with deep learning / M. S. Razali та ін. *IEEE access*. 2022. С. 1. URL: https://doi.org/10.1109/access.2022.3194119 (дата звернення: 02.04.2025).

17. Li J., Ni S., Kao H.-Y. Meet The Truth: Leverage Objective Facts and Subjective Views for Interpretable Rumor Detection. *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, м. Online. Stroudsburg, PA,

USA, 2021. URL: https://doi.org/10.18653/v1/2021.findings-acl.63 (дата звернення: 09.04.2025).

18. Fake News Detection through Graph-based Neural Networks: A Survey / Shuzhi Gong, Richard O. Sinnott, Jianzhong Qi, Cecile Paris, 2023 p. URL: https://doi.org/10.48550/arXiv.2307.12639 (дата звернення 01.04.2025).

19. Adversarial Active Learning Based Heterogeneous Graph Neural Network for Fake News Detection / Y. Ren та ін. *2020 IEEE International Conference on Data Mining (ICDM)*, м. Sorrento, Italy, 17–20 листоп. 2020 р. 2020. URL: https://doi.org/10.1109/icdm50108.2020.00054 (дата звернення: 09.04.2025).

20. KAN: Knowledge-aware Attention Network for Fake News Detection / Y. Dun та ін. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2021. Т. 35, № 1. С. 81–89. URL: https://doi.org/10.1609/aaai.v35i1.16080 (дата звернення: 11.04.2025).

21. Compare to The Knowledge: Graph Neural Fake News Detection with External Knowledge / L. Hu та ін. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing* (Volume 1: Long Papers), м. Online. Stroudsburg, PA, USA, 2021. URL: https://doi.org/10.18653/v1/2021.acl-long.62 (дата звернення: 11.04.2025).

22. Wu K., Yuan X., Ning Y. Incorporating Relational Knowledge in Explainable Fake News Detection. *Advances in Knowledge Discovery and Data Mining*. Cham, 2021. С. 403–415. URL: https://doi.org/10.1007/978-3-030-75768-7_32 (дата звернення: 12.04.2025).

23. Cross-lingual COVID-19 Fake News Detection / J. Du та ін. 2021 *International Conference on Data Mining Workshops (ICDMW)*, м. Auckland, New Zealand, 7–10 груд. 2021 р. 2021. URL: https://doi.org/10.1109/icdmw53433.2021.00110 (дата звернення: 12.04.2025).

24. A Fuzzy Detection System for Rumors through Explainable Adaptive Learning / Z. Guo та ін. *IEEE Transactions on Fuzzy Systems*. 2021. С. 1. URL: https://doi.org/10.1109/tfuzz.2021.3052109 (дата звернення: 12.04.2025).

25. Toward A Multilingual and Multimodal Data Repository for COVID-19 Disinformation / Y. Li та ін. *2020 IEEE International Conference on Big Data (Big Data)*, м. Atlanta, GA, USA, 10–13 груд. 2020 р. URL: https://doi.org/10.1109/bigdata50022.2020.9378472 (дата звернення: 12.04.2025).

26. dEFEND: Explainable Fake News Detection / Kai Shu та ін. 2019 *Association for Computing Machinery*, USA URL: https://doi.org/10.1145/3292500.3330935 (дата звернення: 12.04.2025).

27. Mining Dual Emotion for Fake News Detection / X. Zhang та ін. *WWW '21: The Web Conference 2021*, м. Ljubljana Slovenia. New York, NY, USA, 2021. URL: https://doi.org/10.1145/3442381.3450004 (дата звернення: 13.04.2025).

28. User-Characteristic Enhanced Model for Fake News Detection in Social Media / S. Jiang та ін. *Natural Language Processing and Chinese Computing*. Cham, 2019. С. 634–646. URL: https://doi.org/10.1007/978-3-030-32233-5_49 (дата звернення: 13.04.2025).

29. Lin H., Zhang X., Fu X. A Graph Convolutional Encoder and Decoder Model for Rumor Detection. *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, м. sydney, Australia, 6–9 жовт. 2020 р. 2020. URL: https://doi.org/10.1109/dsaa49011.2020.00043 (дата звернення: 13.04.2025).

30. Wang W. Y. "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (Volume 2: Short Papers), м. Vancouver, Canada. Stroudsburg, PA, USA, 2017. URL: https://doi.org/10.18653/v1/p17-2067 (дата звернення: 13.04.2025).

31. Unsupervised Fake News Detection on Social Media: A Generative Approach / S. Yang та ін. *Proceedings of the AAAI Conference on Artificial Intelligence*. 2019. Т. 33. С. 5644–5651. URL: https://doi.org/10.1609/aaai.v33i01.33015644 (дата звернення: 13.04.2025).

32. Unsupervised rumor detection based on users' behaviors using neural networks / W. Chen та ін. *Pattern Recognition Letters*. 2018. Т. 105. С. 226–233. URL: https://doi.org/10.1016/j.patrec.2017.10.014 (дата звернення: 14.04.2025).

33. BuzzFeed-Webis Fake News Corpus 16. *Webis*. URL: https://webis.de/data/buzzfeed-webis-fake-news-16.html#publications (дата звернення: 20.04.2025).

34. Kaggle. *Kaggle.com* URL: https://www.kaggle.com/datasets/rmisra/politifact-fact-check-dataset (дата звернення: 01.05.2025)

35. Dropbox. *Dropbox.com*. URL: https://www.dropbox.com/scl/fi/flgahafqckxtup2s9eez8/rumdetect2017.zip?dl=0&e=1&file_subpath=/rumor_detection_acl2017/twitter16&rlkey=b7v86v3q1dpvcutxqk0xi7oej (дата звернення: 01.05.2025).

36. PHEME dataset for Rumour Detection and Veracity Classification. figshare. URL: https://figshare.com/articles/dataset/PHEME_dataset_for_Rumour_Detection_and_Veracity_Classification/6392078 (дата звернення: 16.04.2025).

37. Fakeddit. *Fakeddit.* URL: https://fakeddit.netlify.app/ (дата звернення: 01.05.2025).

38. GitHub - EnyanDai/FakeHealth: This repository (FakeHealth) is collected to address challenges in Fake Health News detection. *GitHub.* URL: https://github.com/EnyanDai/FakeHealth (дата звернення: 01.05.2025).

39. GitHub - cuilimeng/CoAID. *GitHub.* URL: https://github.com/cuilimeng/CoAID/tree/master (дата звернення: 01.05.2025).

40. GitHub - Gautamshahi/FakeCovid: FakeCovid- A Multilingual Cross-domain Fact Check News Dataset for COVID-19. *GitHub*. URL: https://github.com/Gautamshahi/FakeCovid/tree/master (дата звернення: 01.05.2025).

41. GitHub - apurvamulay/ReCOVery: A Multimodal Repository for COVID-19 News Credibility Research. *GitHub*. URL: https://github.com/apurvamulay/ReCOVery/tree/master (дата звернення: 01.05.2025).

42. GitHub - bigheiniu/MM-COVID: Cross Linugual COVID-19 Fake News Dataset. *GitHub*. URL: https://github.com/bigheiniu/MM-COVID?tab=readme-ov-file (дата звернення: 01.05.2025).

43. News headlines dataset for sarcasm detection. *Kaggle.com*. URL: https://www.kaggle.com/datasets/rmisra/news-headlines-dataset-for-sarcasm-detection (дата звернення: 01.05.2025)

44. Tweets with sarcasm and irony. *Kaggle.com.* URL: https://www.kaggle.com/datasets/nikhiljohnk/tweets-with-sarcasm-and-irony (дата звернення: 01.05.2025).

45. Scientist S. A. D. Multi-Agent AI Systems:Foundational Concepts and Architectures. *Medium.* URL: https://medium.com/@sahin.samia/multi-agent-ai-systems-foundational-concepts-and-architectures-ece9f8859302 (дата звернення: 02.04.2025).

46. google-bert/bert-base-multilingual-cased · Hugging Face. *Hugging Face – The AI community building the future.* URL: https://huggingface.co/google-bert/bert-base-multilingual-cased (дата звернення: 02.05.2025).

47. J. Devlin, M. Chang, K. Lee, and K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (2019), *North American Chapter of the Association for Computational Linguistics* URL: https://doi.org/10.48550/arXiv.1810.04805 (дата звернення: 03.05.2025).

48. Heterogeneous Graph Attention Network / X. Wang та ін. WWW '19: The Web Conference, м. San Francisco CA USA. New York, NY, USA, 2019. URL: https://doi.org/10.1145/3308558.3313562 (дата звернення: 06.05.2025).

49. tabularisai/multilingual-sentiment-analysis Hugging Face. *Hugging Face – The AI community building the future*. URL: https://huggingface.co/tabularisai/multilingual-sentiment-analysis (дата звернення: 22.05.2025).