Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

ДОСЛІДЖЕННЯ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ ДЛЯ АНАЛІЗУ ТА ГЕНЕРАЦІЇ ХУДОЖНІХ ТЕКСТІВ
(тема)

Виконав:
здобувач 2 року навчання,
групи ІНФм-24-1
Супрун А. Є.
(прізвище, ініціали)

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

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Науковий керівник доц. Творошенко І. С.
(посада, прізвище, ініціали)

Допускається до захисту

Завідувач кафедри інформатики _____ Кобилін О. А.
(підпис) (прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту

Кафедра Інформатики

Рівень вищої освіти другий (магістерський)

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

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Супрун Анні Євгенівні
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження великих мовних моделей для аналізу та генерації художніх текстів

затверджена наказом університету від 14 листопада 2025 року № 1045Ст
2. Термін подання здобувачем роботи до екзаменаційної комісії 22 листопада 2025 р.
3. Вихідні дані до роботи великі мовні моделі GPT-4o, Claude 4 Opus, Gemini 2.5 Pro, літературні джерела щодо застосування великих мовних моделей, програмні засоби для реалізації розробленого методу комбінованого використання великих мовних моделей, набір тестових сценаріїв у вигляді описів персонажів, світу і запитів користувача.

4. Перелік питань, що потрібно опрацювати в роботі
1. Аналіз можливостей великих мовних моделей у задачах аналізу і генерації тексту.
2. Аналіз літературних джерел щодо апробації використання великих мовних моделей у задачах аналізу і генерації тексту і оцінки якості творчої генерації LLM.
3. Формування покрокового алгоритму методу комбінованого використання великих мовних моделей.
4. Візуалізація сформованого покрокових алгоритму.
5. Розроблення програмного застосунку, що надасть змогу тестувати моделі та збирати метрики якості у завданнях аналізу і генерації.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність проблеми застосування великих мовних моделей для аналізу і генерації художніх текстів, об'єкт та мета дослідження, постановка задачі, блок-схеми алгоритму комбінованого методу, ілюстрації роботи програмного застосунку по отриманню відповідей від моделей і підрахунку метрик, графіки порівняння результатів моделей у завданнях аналізу, графіки порівняння результатів моделей у завданнях генерації, графіки порівняння результатів комбінованого методу, перспективи та апробація роботи.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	29.09.2025	
2	Аналіз завдання, підбір літератури	30.09.25-07.10.25	
3	Аналіз літератури з досліджуваної проблеми	08.10.25-14.10.25	
4	Особливості великих мовних моделей	15.10.25-20.10.25	
5	Дослідження великих мовних моделей	21.10.25-27.10.25	
6	Програмна реалізація	28.10.25-05.11.25	
7	Обґрунтування отриманих результатів	06.11.25-11.11.25	
8	Оформлення пояснювальної записки	12.11.25-14.11.25	
9	Перевірка на нормоконтроль	19.11.25-10.12.25	
10	Перевірка на плагіат	20.11.25-10.12.25	
11	Рецензування	21.11.25-10.12.25	
12	Підготовка презентації та доповіді	21.11.25-22.12.25	
13	Занесення роботи в електронний архів	21.11.25-22.12.25	
14	Попередній захист кваліфікаційної роботи	01.12.25-22.12.25	

Дата видачі завдання 29 вересня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ доц. Творошенко І. С.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи: 170 с., 10 табл., 16 рис., 3 дод., 47 джерел.

АВТОМАТИЧНІ МЕТРИКИ, АНАЛІЗ ТЕКСТУ, ВЕЛИКІ МОВНІ МОДЕЛІ, ГЕНЕРАЦІЯ ТЕКСТУ, ЕКСПЕРТНА ОЦІНКА, КОМБІНОВАНИЙ МЕТОД, ЛІТЕРАТУРНА ГЕНЕРАЦІЯ, ОЦІНКА ЯКОСТІ ГЕНЕРАЦІЇ, CLAUDE 4 OPUS, GEMINI 2.5 PRO, GPT-4O, LLM, PROMPT ENGINEERING.

Об’єктом дослідження є великі мовні моделі.

Метою дослідження є порівняння великих мовних моделей у завданнях аналізу і генерації художніх текстів шляхом розробки застосунку, що оцінює результати роботи моделей за системою експертних і автоматичних метрик та реалізує комбінований метод взаємодії кращих з них для поліпшення якості.

Використано моделі GPT-4o, Claude 3 та Gemini 2.5 Pro для аналізу й генерації художніх текстів. Проведено експерименти із трьома сценаріями, у яких порівнювалися результати аналітичного розбору та генерації текстів. Результати візуалізовано у вигляді таблиць і графіків.

Наукова новизна роботи полягає у створенні комбінованого методу використання великих мовних моделей, у якому розділено функції аналізу та генерації між двома різними моделями. Такий підхід підвищує смислову глибину, послідовність і художню виразність текстів.

Взаємозв’язок з іншими роботами полягає у використанні сучасних підходів до оцінки якості роботи LLM, а також у продовженні досліджень у галузі інтелектуальної творчості та автоматизованої літературної генерації.

Рекомендації щодо використання результатів роботи: використання комбінованого методу для створення інструментів для письменників.

У результаті дослідження розроблено застосунок для оцінки якості роботи великих мовних моделей у завданнях аналізу й генерації текстів.

ABSTRACT

Explanatory note to the qualification work: 170 pages, 10 table, 16 figures, 3 appendixes, 47 sources.

AUTOMATIC METRICS, CLAUDE 4 OPUS, COMBINED METHOD, EXPERT EVALUATION, GEMINI 2.5 PRO, GENERATION QUALITY ASSESSMENT, GPT-4O, LARGE LANGUAGE MODELS, LITERARY GENERATION, PROMPT ENGINEERING, TEXT ANALYSIS, TEXT GENERATION.

The object of the research is large language models.

The aim of the research is to compare large language models in the tasks of analyzing and generating literary texts by developing an application that evaluates the results of the models' work using a system of expert and automatic metrics and implements a combined method of interacting the best of them to improve the quality.

The GPT-4o, Claude 3, and Gemini 2.5 Pro models were used to analyze and generate literary texts. Experiments were conducted with three scenarios, in which the results of analytical analysis and text generation were compared. The results are visualized in the form of tables and graphs.

Scientific novelty of the work lies in the creation of a combined method of using large language models, in which the functions of analysis and generation are divided between two different models. This approach increases the semantic depth, consistency and artistic expressiveness of texts.

Interconnection with other works lies in the use of modern approaches to assessing the quality of LLM work and in continuing research in the field of intellectual creativity and automated literary generation.

Recommendations for using the results: creation of tools for writers.

As a result of the research, an application was developed to assess the quality of work of large language models in text analysis and generation tasks.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД – база даних

млн. – мільйон

тис. – тисяча

ШІ – штучний інтелект

AI – Artificial Intelligence (штучний інтелект)

API – Application Programming Interface (інтерфейс програмування застосунків)

BART – Bidirectional and Auto-Regressive Transformer (двонаправлений та авторегресивний трансформер)

BERT – Bidirectional Encoder Representations from Transformers (двоспрямовані кодувальні представлення з трансформерів)

BLEU – Bilingual Evaluation Understudy (автоматична двомовна оцінка)

LLM – Large Language Model (велика мовна модель)

RAG – Retrieval-Augmented Generation (генерація з доповненням через пошук)

RLHF – Reinforcement Learning with Human Feedback (навчання з підкріпленням на основі відгуків людей)

ROUGE – Recall-Oriented Understudy for Gisting Evaluation (орієнтована на повноту оцінка якості узагальнення)

SLM – Small Language Model (мала мовна модель)

SOTA – State of The Art (найвищий рівень загального розвитку)

ВСТУП

У сучасному світі, де технології штучного інтелекту стрімко розвиваються, великі мовні моделі вийшли за межі суто академічних досліджень і знайшли широке застосування в різних галузях. Особливий інтерес становить їхній потенціал у сфері творчості, а саме, для аналізу та генерації художніх текстів.

Традиційні методи літературного аналізу вимагають значних людських ресурсів і часу, а генерація нових творів є виключно прерогативою автора. Завдяки своїй здатності обробляти величезні обсяги інформації, розуміти контекст і продукувати зв’язний текст, великі мовні моделі відкривають нові горизонти для цих процесів. Вони можуть стати інструментами для дослідження глибинних структур літератури, експериментальної генерації сюжетів, описів персонажів і живих діалогів, доповнення внутрішнього бачення авторів, а також для допомоги письменникам у творчому процесі [1].

Наразі існує багато LLM, як пропрієтарних (GPT-4o, Claude 4 Opus, Gemini 2.5 Pro), так і відкритих (LLaMA 3, Mixtral 8x7B), кожна з яких має свої архітектурні особливості та унікальні можливості.

У контексті застосування LLM для художньої творчості важливим залишається питання, наскільки глибоко ці моделі здатні розуміти складні літературні концепції, як-от психологію персонажів, логіку світу, внутрішні конфлікти, і наскільки реалістично вони можуть відтворювати їх у згенерованому тексті.

Основною проблемою генерації художніх текстів є те, що навіть найсучасніші мовні моделі схильні до створення поверхневих чи шаблонних сюжетних ліній, браку емоційної глибини й оригінальності. Моделі добре відтворюють стиль і лексику. Однак, вони часто не здатні повністю відобразити багатошаровість людських переживань, психологічні нюанси персонажів, складні мотиви їхніх дій і глибинні взаємозв’язки між подіями й правилами світу.

LLM можуть втрачати цілісність на довгих відрізках тексту, що призводить до логічних суперечностей у сюжеті чи до плутанини в розвитку теми.

Задачею кваліфікаційної роботи є дослідити та здійснити порівняльний аналіз можливостей великих мовних моделей для аналізу та генерації художніх текстів на основі наданого контексту, а також розробити методику для їхнього ефективного використання.

Для вирішення визначеної проблеми буде проведено аналіз сучасних великих мовних моделей, що демонструють потенціал для роботи з художніми текстами, із особливою увагою до їхніх архітектурних особливостей, що дозволить розробити методику подачі вхідних даних, а також сформувати ефективні запити (промпти) для якісної генерації тексту.

Окрім того, буде реалізовано програмний застосунок, призначений для генерації художніх текстів за допомогою LLM на основі наданих художніх описів за заданим промптом користувача.

Під час експериментального порівняння великих мовних моделей, таких як GPT-4o, Claude 4 Opus і Gemini 2.5 Pro, буде застосована одна й та ж сама задача генерації художнього тексту.

Результати експериментального порівняння будуть оцінені за критеріями зв’язності, відповідності контексту, реалістичності діалогів та психологічної достовірності персонажів. На основі результатів тестування буде проведено порівняльний аналіз, який дозволить виділити переваги та недоліки кожної з LLM, визначено перспективи їхнього подальшого використання у сфері художньої творчості, а також запропоновано оригінальний метод для аналізу та генерації художнього тексту.

Результати дослідження матимуть практичне значення для письменників, сценаристів та розробників інструментів для написання художніх текстів.

Отже, тема кваліфікаційної роботи «Дослідження великих мовних моделей для аналізу та генерації художніх текстів» є надзвичайно актуальною та має як теоретичне, так і практичне значення.

1 АНАЛІЗ ІСНУЮЧИХ МОВНИХ МОДЕЛЕЙ

1.1 Аналіз сучасних великих мовних моделей та приклади їх практичного використання

1.1.1 Загальна характеристика великих мовних моделей

Великі мовні моделі (Large Language Models, LLMs) – це клас штучних нейронних мереж, спеціально розроблених для обробки й генерації природної мови на основі статистичного аналізу великих обсягів текстових даних. LLM не оперують фіксованими правилами, на відміну від традиційних алгоритмів обробки тексту, а навчаються виявляти закономірності у мовних структурах, прогнозувати наступні слова і будувати семантично та синтаксично зв’язні тексти [2].

Термін «велика» вказує на масштаб моделі – кількість параметрів (ваг нейронної мережі), що може коливатися від мільярдів до трильйонів. До прикладу, модель GPT-3 має 175 мільярдів параметрів, GPT-4 – орієнтовно понад трильйон, а сучасні відкриті моделі, такі як LLaMA 3 чи Mistral, мають від 7 до 70 мільярдів параметрів [3]. Значний розмір дозволяє моделі засвоювати складніші шаблони та відтворювати глибші смислові залежності між словами і фразами.

Основні характеристики LLM, котрі визначають їх ефективність:

– кількість параметрів, що впливає на глибину розуміння та генерації;

– обсяг тренувальних даних, що включає терабайти текстів з відкритих інтернет-ресурсів, книг, наукових статей і спеціалізованих текстових зібрань. Великі обсяги підвищують мовне різноманіття даної моделі і вимагають ретельну фільтрацію даних для зменшення упередженості. Наприклад, Gemini 1.5 від Google навчалася на багатомовному мультимодальному наборі, що включав текст, зображення та код [4];

– довжина контекстного вікна, що визначає, який обсяг попереднього тексту модель може враховувати при генерації. До прикладу, GPT-4 Turbo

підтримує контекст до 128 тис. токенів, Claude 3.5 Sonnet – до 200 тис., Gemini 1.5 Pro – до 1 млн. токенів;

– архітектурні оптимізації – використання механізму multi-head attention, нормалізації й оптимізованих алгоритмів навчання, позиційного кодування, використання ефективних оптимізаторів (AdamW, Lion) і прискорених бібліотек обчислень (FlashAttention).

Сьогодні LLM знаходять застосування у багатьох сферах: у чат-ботах та автоматичному перекладі, генерації коду, аналізі даних і створенні художніх творів. Важливо, що у творчих завданнях вони здатні імітувати різні стилі письма, відтворювати логіку персонажів, «розуміти» правила світу та підтримувати цілісність сюжету [5]. Втім, для досягнення високої точності в таких завданнях необхідне ретельне налаштування й інколи комбінація декількох моделей, кожна з яких виконує свою спеціалізовану роль.

Великі мовні моделі є не простими інструментами для обробки мови, а універсальними системами штучного інтелекту (ШІ), здатними до генерації змістовного тексту, аналізу контексту, адаптації до конкретних завдань, створення нових ідей. Їхня ефективність безпосередньо залежить від архітектури, кількості параметрів, обсягу даних і якості навчання. Це робить їх дослідження актуальним у сучасній науці та практиці.

Найбільш відомі LLM у 2024 та 2025 роках, що успішно використані:

– GPT-4o (OpenAI) – універсальна модель з високою якістю генерації тексту й багатомовною підтримкою;

– Claude 4 Opus (Anthropic) – модель, оптимізована для довгих діалогів і логічних міркувань, із розширеним контекстним вікном;

– Gemini 2.5 Pro / Gemini 2.5 Flash (Google DeepMind) – мультимодальна LLM, здатна аналізувати текст, зображення, аудіо й відео;

– LLaMA 3 (Meta AI) – відкрита модель, популярна серед дослідників через можливість локального розгортання;

– DeepSeek-V3 (DeepSeek AI) – китайська LLM з оптимізованим співвідношенням якості до обчислювальних витрат;

– Mistral Large (Mistral AI) – компактна, але потужна модель з відкритою ліцензією, ефективна для завдань на обмежених ресурсах.

1.1.2 Архітектурні особливості сучасних великих мовних моделей

Основою більшості сучасних LLM є архітектура Transformer, яка запропонована в роботі [6]. Найголовнішим елементом трансформера є механізм attention (уваги), котрий дозволяє моделі оцінювати важливість кожного слова в контексті речення чи навіть всього документа. Він дає змогу ефективно працювати з довгими залежностями та контекстом, що є критично важливим для генерації зв'язного тексту.

Механізм attention – це компонент нейронних мереж, котрий дозволяє моделі вибірково зосереджуватися на релевантних частинах вхідних даних під час обробки. У контексті обробки природної мови це означає, що кожен токен у вхідній послідовності враховує взаємозв'язки з іншими токенами, формуючи більш точне та контекстуально залежне подання. Даний механізм формалізується через вагові коефіцієнти, що застосовуються для зважування внеску окремих токенів у формування вихідного подання.

Механізм attention визначається як

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right), \quad (1.1)$$

де Q – матриця запитів;

K^T – транспонована матриця ключів;

V – матриця значень;

d_k – розмірність векторів ключів.

Multi-head attention – це вдосконалений варіант механізму attention, що застосовує кілька незалежних каналів уваги (heads) паралельно.

Кожен канал уваги виконує власний обчислювальний процес attention і може зосереджуватися на різних аспектах вхідної інформації: один – на синтаксичних зв’язках між словами, інший – на семантичних асоціаціях або на довготривалих залежностях у тексті тощо.

Отримані результати з усіх каналів об’єднуються і проходять через лінійний шар нейронної мережі. Це дає можливість їй одночасно враховувати кілька різних типів взаємозв’язків і будувати більш інформативне багатовимірне подання для подальшої обробки.

Тренування великих мовних моделей, зазвичай, відбувається у два етапи:

Етап 1. Попереднє тренування (pre-training) – модель навчається на великих корпусах текстів (книги, статті, вебсторінки) стосовно завдання передбачення наступного слова чи заповнення пропуску в тексті.

Етап 2. Донавчання (fine-tuning) – модель додатково навчається на вузькоспеціалізованих даних чи з використанням інструкцій, щоби підвищити точність виконання конкретних завдань.

Деякі LLM додатково проходять етап Reinforcement Learning with Human Feedback – навчання з підкріпленням за допомогою зворотного зв’язку від людини. Такий підхід покращує відповідність відповідей моделі людським очікуванням, зменшиує кількість некоректних чи шкідливих результатів.

Архітектура трансформера складається з модулів кодування (encoder) та генерації (decoder), котрі можуть комбінуватися залежно від завдань:

– encoder-only (BERT) – моделі, що складаються тільки з модулів кодування, вони використовуються для завдань аналізу тексту: класифікації, вилучення сутностей, семантичного пошуку;

– decoder-only (GPT-3/4, LLaMA) – моделі, що складаються лише з модулів генерації. Вони використовуються переважно для генерації тексту, бо кожний новий токен прогнозується на основі попередніх;

– encoder-decoder (T5, BART) – моделі, які поєднують обидві частини. Застосовуються у перекладі, перефразуванні, сумаризації, коли необхідне перетворення одного тексту на інший.

Модуль кодування – це частина моделі, що перетворює вхідний текст у числові вектори, зберігаючи його зміст і структуру. Кожен токен (слово, частина слова чи символ) отримує вектор-ембеддінг, який у кількох шарах attention та нейронних блоках збагачується контекстом: модель враховує взаємозв'язки між усіма словами в реченні або абзаці. Так формується контекстуалізоване подання, придатне для аналізу, класифікації чи пошуку смислових зв'язків. До прикладу, BERT, що використовує лише модулі кодування й чудово працює для завдань аналізу тексту (класифікація, емоції, сутності).

Модуль генерації – це частина моделі, що відповідає за генерацію тексту. Він бере внутрішні подання (із модуля кодування або з попередніх токенів у випадку decoder-only) і по черзі передбачає наступне слово (autoregressive generation). Кожний новий токен враховує все згенероване раніше. Механізм multi-head attention дає декодеру фокусуватись на різних аспектах тексту одночасно, що покращує якість результату. Завдяки цьому, такі моделі застосовують для перекладу, сумаризації, діалогів та творчої генерації історій.

Схему трансформера encoder-decoder зображено на рисунку 1.1 [6].

Вхідний текст подається на послідовність модулів кодування, де формується контекстуалізоване подання кожного токена.

Модулі генерації використовують ці подання для генерації вихідного тексту, послідовно прогнозуючи наступні токени. На схемі також показано механізми self-attention та multi-head attention, котрі дозволяють моделі одночасно оцінювати різні аспекти зв'язків між словами, що забезпечує ефективну роботу із довгими залежностями та контекстом.

Великі мовні моделі можуть працювати в різних режимах залежно від наявності додаткових прикладів чи налаштування під конкретне завдання:

– Zero-shot learning: модель виконує завдання без жодного додаткового навчання, спираючись тільки на інструкцію користувача. Прикладом є GPT-4, яка вміє одразу перекладати текст з української на японську, не отримуючи прикладів;

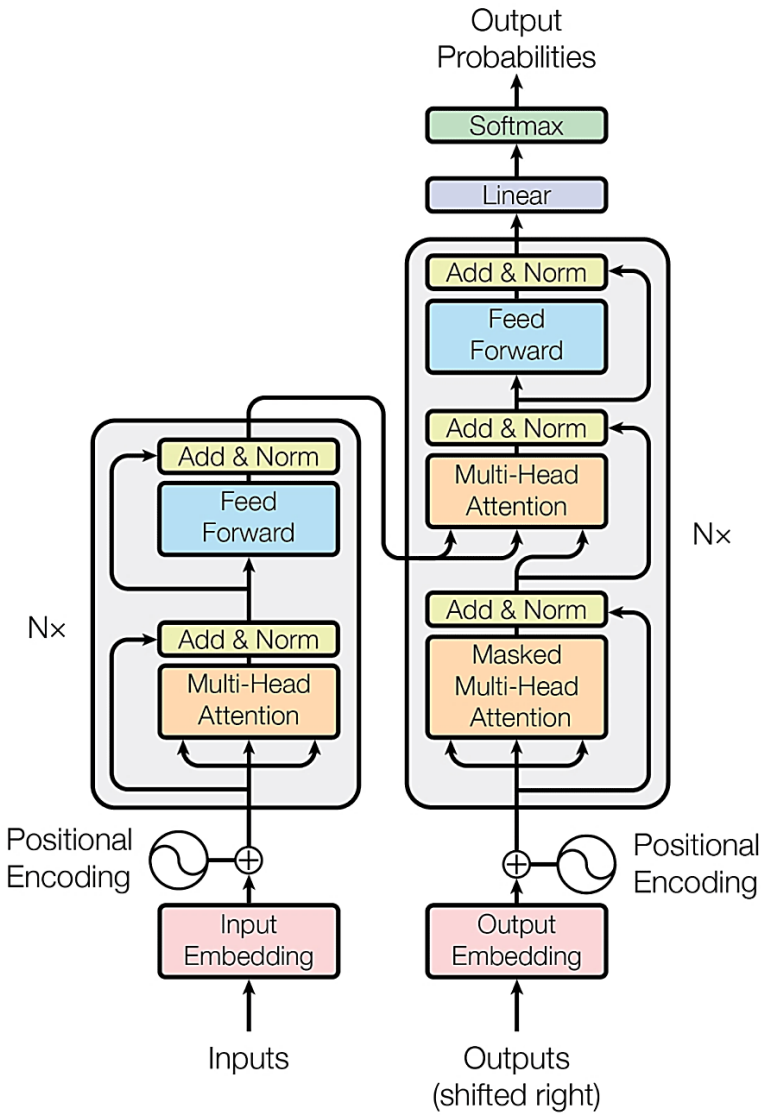

Рисунок 1.1 – Архітектура трансформера (encoder-decoder)

– Few-shot learning: у промпті подаються декілька прикладів виконання завдання, потім модель генерує відповідь у подібному форматі, це підвищує точність у вузькоспеціалізованих завданнях;

– Fine-tuning: додаткове навчання моделі на спеціалізованому наборі даних для досягнення максимальної точності у певній області (наприклад, літературна стилізація або медична термінологія). Fine-tuning може бути повним (із оновленням усіх параметрів) або частковим (наприклад, через Low-Rank Adaptation).

Сучасні LLM розвиваються в напрямках збільшення контекстного вікна, вироблення ефективніших механізмів attention, мультимодальності.

Моделі, такі як Claude 3 чи GPT-4 Turbo, можуть обробляти до 200 тис. токенів контексту. Це дає змогу аналізувати цілі книги або великі кодові бази. GPT-4, Gemini 1.5 мають в собі інтеграцію обробки тексту, зображень, аудіо, відео в єдиній архітектурі. Завдяки цим особливостям новітні LLM можуть ефективно працювати з текстом як для аналізу, так і для генерації нових сюжетів, персонажів і діалогів.

1.1.3 Сфери практичного використання великих мовних моделей

Однією із найпопулярніших сфер застосування LLM стали чат-боти та інтерактивні асистенти. Якщо ранні системи базувалися на простих правилах, то сучасні (ChatGPT, Claude, Gemini) демонструють якісно інший рівень: ведуть діалог, розуміють складні інструкції, зберігають контекст та адаптуються стилем під користувача. На практиці зазначені можливості використовуються в клієнтській підтримці та освіті, де LLM зменшують навантаження на персонал, підвищуючи ефективність. За даними Microsoft, інтеграція LLM у call-центри підвищує продуктивність операторів на 14% [7].

Важлива сфера – програмування. GitHub Copilot (на базі OpenAI Codex) та Code LLaMA допомагають розробникам автодоповненням, рефакторингом та пошуком помилок. За офіційними даними GitHub, використання інструменту ШІ Copilot скорочує час написання коду приблизно на 55%, що змінює сам підхід до розроблення.

У бізнесі LLM використовуються для аналізу текстів, резюмування, перекладу та автоматизації документів. Перелічені можливості актуальні для юридичних, фінансових компаній і глобалізації бізнесу, де автоматичний переклад (Google, DeepL) наближається до рівня професійних.

Окрема ніша – творчість: LLM генерують оповідання, сценарії, діалоги у відеоіграх, вони допомагають авторам долати «творчий ступор» і створюють динамічний контент.

Деякі студії вже експериментують із генерацією унікальних діалогів у режимі реального часу [8].

Варто очікувати, що в найближчому майбутньому LLM стануть ще більш інтегрованими у повсякденне життя суспільства – від автоматичного супроводу навчального процесу до персоналізованих цифрових асистентів, що діятимуть на перетині аналізу, генерації та розуміння людської мови.

1.1.4 Використання великих мовних моделей у сфері літературної творчості

Великі мовні моделі відкривають нові можливості у створенні художніх текстів, стаючи інструментом для письменників, сценаристів та ігрових дизайнерів. На відміну від алгоритмів із жорсткими шаблонами, LLM враховують контекст, стиль, психологію персонажів, здатні розвивати сюжет у межах жанру чи художньої концепції.

Важливою перевагою LLM є послідовність і достовірність оповіді: модель може відтворювати логіку персонажів і їхні емоційні реакції, забезпечуючи цілісність тексту. Модель також легко адаптується до різних жанрів і стилів, від класичної прози до сучасних експериментів, що робить її зручним інструментом для редагування та пошуку нових сюжетних рішень.

Великий інтерес викликає інтерактивна генерація, коли автор задає промпти й отримує кілька варіантів розвитку сцени. Це використовується у створенні інтерактивних історій та відеоігор з динамічними діалогами.

Серед прикладів програм – Sudowrite, що допомагає долати творчу кризу, та NovelAI, яка спеціалізується на прозі та створенні сюжетних арок. Окрім генерації, LLM застосовуються для аналізу літератури: визначення емоційного тону, тематики, стилістики чи навіть упізнавання авторського стилю.

Варто відзначити, що інтеграція LLM у творчий процес не є спробою замінити автора.

Радше йдеться про створення нової форми співпраці, бо модель бере на себе рутинні або технічні завдання (генерація діалогів, описів, варіантів розвитку подій), тоді як письменник зберігає контроль над змістом, стилем і художнім задумом. У цьому сенсі LLM можна розглядати як інтелектуальний «інструмент натхнення», що розширює межі людської уяви.

Сучасні приклади використання великих мовних моделей у літературній творчості демонструють їхню здатність працювати у ролі генератора текстів й аналітичного інструменту.

1.1.5 Обґрунтування вибору великих мовних моделей для дослідження

У межах дослідження обрано три провідні великі мовні моделі: GPT-4o (OpenAI), Claude 4 Opus (Anthropic) та Gemini 2.5 Pro (Google DeepMind). Вибір цих моделей обумовлений їхніми передовими можливостями у сфері генеративного штучного інтелекту та здатністю до комплексного аналізу контексту й поведінки персонажів.

GPT-4o є мультимодальною моделлю, здатною обробляти текст, зображення й аудіо, що дає можливість глибоко інтегрувати надані користувачем описи світу і персонажів. Вона відзначається високою здатністю до розуміння внутрішніх мотивів і психології героїв, а також до генерації текстів, що відповідають заданому стилю та жанру. Ця модель демонструє найвищу ефективність у задачах художньої творчості, де важлива тонка передача характеру персонажів і логіки сюжету (OpenAI, 2023).

Claude 4 Opus, представлена компанією Anthropic у 2025 році, спеціалізується на багатокроковому міркуванні та структурованому аналізі інформації. Модель добре справляється з розумінням загальної логіки сцен і взаємодій між персонажами, проте менш тонко передає емоційні нюанси. Є в дослідженні як модель порівняння, що дає оцінити відмінності між структурованим підходом і художньою генерацією тексту (Anthropic, 2025).

Gemini 2.5 Pro від Google DeepMind поєднує глибину багатокрокового міркування з можливістю інтеграції різних типів даних. Модель добре підходить для генерації складних сценаріїв і сюжетів, де важливо передбачати можливі взаємодії між персонажами та розвитком подій. Її використання дозволяє порівнювати ефективність підходів різних розробників великих мовних моделей у задачах творчого письма (Google Cloud, 2025).

Обрані моделі репрезентують різні стратегії розвитку великих мовних моделей, дозволяючи провести всебічний аналіз їхніх можливостей. GPT-4o забезпечує найвищу точність у художній генерації та психологічному аналізі персонажів, Claude 4 Opus дає змогу оцінити структурований і логічний підхід, Gemini 2.5 Pro показує потенціал для створення складних сюжетів.

Поєднання зазначених моделей дає можливість не тільки оцінити сильні та слабкі сторони кожної з них, а й розробити оптимальні методики для аналізу та генерації художніх текстів у межах поставлених завдань.

1.2 Аналіз літературних джерел щодо апробації результатів застосування існуючих великих мовних моделей

Для забезпечення повного розуміння проблематики дослідження проведено аналіз наукових джерел. Представлені результати стосуються використання LLM для створення та аналізу художніх текстів, оцінки наративних і креативних здібностей моделей.

У роботі [9] проаналізовано здатність сучасних великих мовних моделей до англомовного художнього письма шляхом порівняння їхніх текстів із творами людей. Завданням було створення епічного опису бою персонажа з птеродактилем, що дозволило оцінити оригінальність, стиль, зв’язність, гумор і мовну плавність. Для дослідження використано такі моделі: ChatGPT (GPT-4), GPT-3.5, Claude 1.2, Gemini 1.5, Alpaca, а також інші моделі, налаштовані на інструкційну взаємодію. Результати показали, що деякі комерційні LLM здатні

дорівнювати або перевищувати людські навички написання художніх творів у більшості аспектів, тоді як відкриті моделі істотно відстають. Сильними сторонами є чітка методика оцінювання та використання людської експертної оцінки, а також постановка завдання, яке неможливо відтворити з навчальних даних. Слабкі сторони полягають у зосередженості лише на англомовному контексті, поверхневому розгляді психології персонажів і відсутності аналізу новітніх моделей (Gemini 1.5 Pro чи Claude 3). Джерело є корисним своєю методологією оцінювання, яка може бути застосована для аналізу якості створених LLM художніх текстів, демонструє різницю між комерційними та відкритими моделями. Пропонується цікава постановка завдання, котра може бути використана як приклад у власному експериментальному дослідженні.

У роботі [10] автори дослідили процес співтворчості людини та великих мовних моделей під час підготовки до написання тексту – як користувачі взаємодіють із LLM для генерації ідей, планування і створення текстів, визначення ролі й ініціатив моделей у процесі. Результати показали наявність трьох ітеративних етапів співтворчості (ідея, освітлення і реалізація), при яких домінуючу роль зберігає людина, а ініціатива між людиною і LLM змінюється. Зазначено різні сприйняття користувачів щодо можливостей моделей. Детально проаналізована динаміка взаємодії людини і LLM із використанням реальних сценаріїв творчої роботи. Застосовується невелика вибірка та є фокус лише на підготовчому етапі письма без оцінки якості кінцевих художніх текстів. Робота надає приклад методології оцінки взаємодії користувача і LLM під час творчого процесу, дозволяє зрозуміти, як моделі допомагають у формуванні ідей для сюжетів, пропонує структуру співтворчого процесу, яку можна врахувати при експериментальному порівнянні різних моделей у художньому письмі.

У роботі [11] досліджено здатність LLM демонструвати дивергентне мислення і семантичну різноманітність у порівнянні з людьми. За допомогою Divergent Association Task і великої вибірки із 100 тис. учасників автори показали, що сучасні моделі можуть перевершувати середній рівень людей і наближатися до творчого письма, але не досягають рівня найкреативніших

авторів. Сильними сторонами дослідження є систематичне порівняння людей і моделей, використання об’єктивних метрик семантичної різноманітності. Обмеження полягає у відсутності оцінки сюжетної логіки і глибини персонажів.

У роботі [12] досліджено творчу здатність LLM у короткому художньому письмі порівняно з людьми, використовуючи завдання на створення історій за п’ятьма ключовими словами. Оцінку проводили через автоматизовані метрики (новизна, несподіваність, різноманітність, лінгвістична складність) і людські оцінки (експертів і неекспертів). Результати показали, що LLM генерують стилістично складні тексти, але поступаються людям у новизні й різноманітності. Водночас LLM і неексперти часто оцінюють машинні тексти як більш креативні, ніж людські, що вказує на різницю в сприйнятті творчості. Джерело показує, які аспекти творчості LLM доцільно оцінювати автоматично та людськими експертами, а результати можна використовувати для порівняння різних моделей у генерації художніх текстів і для розроблення критеріїв оцінки відповідності характерів і сюжету створеним сценам.

Дослідження [13] показує, що ChatGPT можна використовувати як інструмент підтримки творчого письма. Автори зосередилися на одній детальній ситуації, застосувавши методику багатоголосного промптингу: модель одночасно виконувала роль автора й критика, працюючи з описом сцени, елементами сюжету, текстовими прикладами й аналізом зворотного зв’язку. Основним висновком стало те, що рівень витонченості та складності тексту прямо залежить від структури і якості промпту. До сильних аспектів роботи належить поєднання літературного критичного аналізу та методів оцінки творчості LLM, а також ідея багатоголосного промптингу. Обмеження полягає в дослідженні лише одного кейсу і відсутності порівняння з іншими моделями. Результати цього дослідження можна використати для розроблення ефективних промптів і стратегій взаємодії з LLM, що дозволить краще оцінювати здатність моделі відтворювати стиль, атмосферу і характер персонажів у художньому тексті.

У роботі [14] досліджували здатність до творчого письма трьох моделей: BART-large (тонко налаштована SLM), GPT-3.5 і GPT-4o, порівнюючи їх із реальними авторами. Експеримент складався із оцінки коротких історій людьми за граматикою, релевантністю, креативністю і привабливістю, а також якісного лінгвістичного аналізу текстів. Результати показали, що BART-large перевершила людей із середнім рівнем майстерності в написанні художніх текстів у загальній оцінці, хоча GPT-4o демонструвала високу когерентність і менше кліше та була більш передбачуваною і рідше створювала несподівані асоціації.

У статті [15] оцінюється робота GPT-4, Claude 2.1 та LLaMA 2 70B у підсумовуванні коротких оповідань, які спеціально бралися від авторів, щоб уникнути збігу із даними тренування. Дослідження показало, що всі моделі робили помилки у понад 50% випадків, зокрема, у сфері точності, інтерпретації підтексту й передачі деталей. Автори також довели, що автоматичні метрики і навіть самооцінки моделей не корелюють із якістю, яку визначають самі письменники. Залучення авторів як експертних оцінювачів, що робить результати особливо надійними. Ця стаття цінна тим, що підкреслює обмеження LLM у розумінні підтексту та складних наративних структур, що безпосередньо стосується завдання аналізу мотивацій і характерів персонажів у художніх текстах.

У роботі [16] запропоновано нову методику оцінки креативності LLM на основі модифікованих тестів Торренса. Автори протестували низку моделей, серед яких GPT-4, Claude 2, Gemini-1.5 Pro та LLaMA 2 70B, застосовуючи 700 завдань і чотири критерії: гнучкість, оригінальність, плавність і деталізацію. Результати показали, що моделі добре справляються із розгорнутістю і варіативністю відповідей, але слабкіші в оригінальності. Важливим спостереженням є те, що рольові сценарії й правильно побудовані промпти помітно підвищують креативність, а співпраця кількох моделей допомагає компенсувати їхні обмеження. У роботі використовується велика кількість завдань і багатокритеріальна оцінка. Ця робота важлива тим, що

підтверджує ідею комбінування кількох моделей для досягнення більш оригінальних результатів і демонструє як структурування промптів впливає на якість текстів.

У роботі [17] запропоновано автоматизований підхід до оцінки креативності текстів LLM на основі адаптації тесту Торренса (TTCW). Метод передбачає порівняння зразків, згенерованих моделлю, із високоякісними еталонами за допомогою шкали Лайкерта. Результати показали значне підвищення узгодженості із людськими оцінками – до 0,75 точності у парних порівняннях (+15%). Це відкриває можливість інтеграції подібних метрик у дослідження, де важлива об’єктивна оцінка художньої цінності й креативності текстів, зокрема, у роботі над аналізом літературних фрагментів.

У статті [18] проведено порівняння Gemini та ChatGPT, зосереджуючись на архітектурних особливостях, продуктивності й можливостях моделей у різних сферах, від освіти і медицини до фінансів і розваг. Автори оцінювали зв’язність відповідей, точність, швидкодію та масштабованість, показавши, що Gemini вирізняється інноваційними підходами до навчання, а ChatGPT – стабільністю в довготривалих діалогах. Також розглянуто архітектурні відмінності, вплив методів тренування й етичні аспекти взаємодії з користувачем. Сильними сторонами роботи є комплексне порівняння моделей, аналіз продуктивності й архітектури, практичні сценарії застосування, але немає специфічної оцінки творчих чи художніх завдань. Стаття дозволяє зрозуміти, як архітектурні й функціональні особливості моделей впливають на здатність відтворювати контекст, інтенції персонажів та зв’язність у художньому тексті, а також обґрунтовувати комбіноване використання Gemini і ChatGPT для підвищення якості генерації.

У статті [19] досліджуються мовні здібності Google Gemini Pro у порівнянні з GPT-3.5 Turbo на 10 різних завданнях, включно з логічним мисленням, знаннями, математикою, перекладом, генерацією коду та виконанням інструкцій. Результати показали, що Gemini Pro за точністю трохи поступається GPT-3.5 Turbo, особливо в задачах з довгими числами, чутливістю

до порядку варіантів у тестах та через агресивне фільтрування контенту. Gemini демонструє сильні сторони в роботі з неанглійськими мовами й обробці довгих і складних логічних ланцюгів. Проведене порівняння моделей на великому наборі завдань, застосована прозора методика, деталізований аналіз обмежень і переваг Gemini, але оцінка не охоплює творчі та художні аспекти текстів. Стаття показує сильні й слабкі сторони Gemini в мовних задачах і допомагає обґрунтувати її використання в комбінації з іншими LLM для генерації художніх текстів, де важлива робота з довгими контекстами, логікою персонажів і багатомовністю.

Джерело [20] досліджує фактичність довгих текстів, згенерованих LLM, зокрема GPT-4, Gemini 1.5 Pro, Claude 3 Opus, Llama 3 70B і Mistral. Автори показують, що точність поданої інформації знижується в пізніших реченнях, а кількість непідтверджених тверджень зростає. Крім того, вони аналізують здатність моделей оцінювати власні відповіді через метрики Self-Known (правильні твердження, які модель визнає правильними) та Self-Unknown (непідтверджені твердження, які модель визнає неправильними). Навіть просунуті моделі, як GPT-4 і Gemini 1.5 Pro, демонструють обмежену здатність до самоперевірки, а високе значення метрики Self-Known корелює з кращою фактичністю, тоді як високе значення метрики Self-Unknown – зі зниженням фактичності. Дослідження показує обмеження моделей у підтримці точності та достовірності інформації в довгих сюжетних текстах, що допоможе обирати стратегії контролю логіки та фактології в згенерованих сценах і діалогах між персонажами.

У дослідженні [21] проаналізовано здатність GPT-4 емулювати стиль письменника Г. Ф. Лавкрафта. Моделі надавали промпти зі зразками текстів письменника для формування тексту в характерному жанрі жахів. Для оцінки ефективності провели опитування студентів, в котрому респонденти не змогли точно відрізнити згенеровані тексти від оригінальних. У роботі демонструються креативність GPT-4, успішне використання емпіричних методів оцінки і докладний опис архітектури моделі. Мінусами є обмежена вибірка

респондентів і фокус лише на одному авторі й жанрі, а залежність від обраних промптів створює ризик неповторюваності ефекту для інших стилів. Дослідження показує ефективність промпт-інженерії для керування стилем і змістом тексту, дає приклад організації оцінки якості за допомогою експертів і звичайних читачів.

Отже, аналіз низки літературних джерел, що стосуються застосування великих мовних моделей у створенні й оцінці художніх текстів показав, що сучасні LLM здатні генерувати сюжетні ходи, діалоги й ідеї для творчих текстів, а також слугують ефективним інструментом для підтримки письменників на етапі написання чернетки і повноцінного твору. Водночас вони мають суттєві обмеження: недостатню індивідуалізацію стилю, схильність до передбачуваних сюжетів, обмежену здатність створювати глибокі внутрішні конфлікти та багаторівневі наративні структури.

Ці результати підкреслюють актуальність подальших досліджень у сфері підвищення якості художньої генерації, розробки методів співтворчості людини й ШІ, а також створення підходів, що дозволяють комбінувати сильні сторони різних моделей.

1.3 Постановка задачі дослідження

Таким чином, аналіз і генерація художніх текстів за допомогою великих мовних моделей є актуальним завданням. Прийнято рішення щодо проведення порівняльного тестування різних LLM на якість аналізу та генерації художніх текстів із урахуванням психології персонажів, їх мотивацій і відповідності заданій локації.

Об’єктом дослідження є великі мовні моделі.

Метою дослідження є порівняння великих мовних моделей у завданнях аналізу і генерації художніх текстів шляхом розробки застосунку, що оцінює

результати роботи моделей за системою експертних і автоматичних метрик та реалізує комбінований метод взаємодії кращих з них для поліпшення якості.

Прийнято рішення щодо розроблення програмного застосунку, в якому буде реалізована генерація художніх текстів великими мовними моделями на основі промптів користувача, що дозволить здійснити порівняння обраних LLM (GPT-4o, Claude 4 Opus, Gemini 2.5 Pro).

Для досягнення мети необхідно вирішити такі завдання:

− провести аналіз сучасних LLM, що використовуються для генерації й аналізу художніх текстів;

− провести аналіз літературних джерел щодо апробації результатів застосування існуючих великих мовних моделей;

− розробити методику тестування LLM на здатність аналізувати характеристики персонажів, їх мотивації й взаємодії в заданих сюжетних ситуаціях та генерувати якісні художні тексти;

− розробити програмний застосунок, який забезпечить взаємодію користувача із обраними LLM, введення промптів і отримання результатів генерації тексту;

− провести експериментальне порівняння моделей GPT-4o, Claude 4 Opus і Gemini 2.5 Pro за якістю аналізу та генерації художніх текстів;

− проаналізувати результати тестування, визначити сильні та слабкі сторони кожної моделі;

− запропонувати й реалізувати власний підхід (комбіновану систему), що поєднує кілька моделей для підвищення якості генерації художніх текстів;

− оцінити ефективність запропонованого підходу та порівняти його з результатами окремих LLM.

2 ОСОБЛИВОСТІ ВИБРАНИХ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ ДЛЯ АНАЛІЗУ ТА ГЕНЕРАЦІЇ ХУДОЖНІХ ТЕКСТІВ

2.1 Велика мовна модель GPT-4o

Модель GPT-4o (де «o» означає «omni», тобто всеохоплююча) від OpenAI репрезентує філософію SOTA (State-of-the-Art) та універсальності, ставлячи за мету найвищу продуктивність у всіх доменах, включаючи складну творчу генерацію. Її архітектурна перевага полягає в досконалій інтеграції обробки різних модальностей (текст, аудіо, зір) в єдиній нейронній мережі. Хоча для літературної роботи основною є текстова генерація, ця нативна мультимодальність свідчить про глибоку здатність моделі зіставляти різнорідні описи: наприклад, вона ефективно зв’язує візуальні елементи світу чи емоційний опис обличчя персонажа (які письменник міг би надати як вхідні дані) з логікою його подальшої поведінки.

Головна цінність GPT-4o у художній творчості полягає в її стилістичній гнучкості та якості прози. Завдяки тренуванню на колосальному, високоякісному наборі даних, модель здатна імітувати широкий спектр літературних стилів – від класичної готичної прози до лаконічного сучасного діалогу. Вона є ефективною в генерації емоційно насиченого тексту, роблячи себе ідеальним кандидатом на роль генератора фінального художнього тексту в дослідницьких сценаріях, де важлива тонка передача стилю [22].

Водночас, її універсальність може бути й потенційною слабкістю. Моделі, які налаштовані на «правильні» й загальноприйнятні відповіді, можуть іноді втрачати унікальність чи гостроту характеру, якщо письменник не надасть достатньо чітких інструкцій щодо недосконалостей чи суперечливих мотивацій персонажів. Хоч GPT-4o чудово справляється з розумінням контексту, її внутрішня логіка міркування може бути менш прозорою, ніж у моделей, оптимізованих спеціально для багатокрокового логічного аналізу [3].

2.2 Велика мовна модель Claude 4 Opus

Claude 4 Opus від Anthropic є втіленням філософії безпеки, послідовності та глибокого контекстуального міркування. На відміну від універсального підходу OpenAI, Anthropic фокусується на принципах Constitutional AI – навчання моделі з використанням набору конституційних принципів, що сприяє генеруванню більш безпечних, чесних та менш упереджених результатів. У контексті художнього тексту це означає, що Claude часто краще підходить для ролі логічного аналітика, який має глибоко розуміти і відтворювати внутрішні правила світу й мотивації персонажів.

Технічна перевага Claude 4 Opus – це її розширене контекстне вікно, яке традиційно є одним із найбільших на ринку (близько 200 тис. токенів). Ця особливість є важливою для підтримки цілісності великого сюжету. Якщо автор надає десятки сторінок опису світу й деталізовані біографії персонажів, Claude 4 Opus здатна утримувати всі ці деталі в активному контексті одночасно. Це мінімізує ризик «забування» важливих сюжетних або психологічних деталей при генерації наступної сцени [23, 24].

Крім того, Claude 4 Opus часто демонструє видатну здатність до багатокрокового міркування (Chain-of-Thought). При отриманні запиту, що вимагає узгодження дії з описами, модель ефективно планує внутрішній хід логічного аналізу. Наприклад, в контексті художньої літератури, при необхідності генерації поведінки персонажа в емоційно напруженій ситуації, модель може послідовно виконати такі кроки:

Крок 1. Ідентифікація ключової мотивації персонажа (наприклад, прагнення до самотності, згідно з вхідними даними).

Крок 2. Оцінка зовнішнього тригера (наприклад, спроба іншого персонажа до зближення).

Крок 3. Прогнозування реакції, що відповідає внутрішній логіці та характеру (наприклад, відмова через відчуженість, а не агресію).

Ця здатність до декомпозиції складного завдання на послідовні логічні кроки може визначити Claude як ідеального кандидата на роль аналітика психології в архітектурах із розділеним функціоналом моделей.

Потенційним недоліком є її філософія безпеки, котра може призвести до «обережної» генерації, де унікальні чи дещо «шкідливі» (але художньо виправдані) дії персонажів можуть бути згладжені, якщо це не буде чітко прописано в промпті.

2.3 Велика мовна модель Gemini 2.5 Pro

Gemini 2.5 Pro від Google DeepMind поєднує високу ефективність в міркуванні з нативною мультимодальністю і безпрецедентно великим контекстним вікном до 1 млн. токенів. Філософія Google полягає в створенні моделі, яка здатна глибоко інтегрувати розрізнені дані та ефективно масштабуватись до завдань, що вимагають аналізу величезних масивів інформації.

Контекстне вікно у 1 млн. токенів – це архітектурна перевага, яка може змінити підхід до написання великих текстів. Теоретично, це дає моделі можливість аналізувати й генерувати текст у контексті цілого роману, що є критично важливим для художньої цілісності. Письменник може надати повну чернетку роману, і Gemini 2.5 Pro, використовуючи внутрішній механізм RAG, може швидко знаходити і застосовувати потрібні деталі зі світу чи біографій персонажів [25–27].

У контексті дослідження, Gemini 2.5 Pro може бути оцінена як хороший інструмент для аналізу логіки світу. Вона має найбільший потенціал для обробки та інтеграції усіх наданих користувачем даних (описів, правил, історії, тощо). Її можливості багатокрокового міркування дозволяють їй виступати як ефективний «планувальник» сцени, створюючи детальні сценарії взаємодії між персонажами.

Головним ризиком Gemini (особливо в ранніх версіях, що потребує перевірки) може бути якість генерації українською мовою та її стилістична витонченість порівняно з GPT-4o, яка довше домінувала на ринку. Хоча модель є високоякісною, важливо емпірично перевірити, чи не поступається її українська проза конкурентам, особливо коли мова йде про тонку передачу емоцій і художній стиль [4, 28].

2.4 Аналіз ключових відмінностей та спільних рис обраних великих мовних моделей

Описані моделі мають низку фундаментальних спільних рис, які визначають їхню ефективність у складних завданнях, зокрема, у сфері художньої творчості:

– усі три моделі побудовані на основі архітектури Transformer з використанням механізму multi-head attention. Цей підхід забезпечує здатність до довготривалих залежностей і ефективного контекстуального розуміння, що є необхідним для створення зв’язного та логічно послідовного художнього тексту;

– кожна з моделей є мультимодальною, що дає їм інтегрувати інформацію з різних джерел. Це дуже важливо для аналізу художніх текстів, де опис світу та персонажів може бути представлений у різних форматах (текст, схеми, зображення);

– навчання з підкріпленням (RLHF/Constitutional AI). Усі моделі пройшли етап додаткового налаштування за допомогою зворотного зв’язку (як від людей, так і на основі принципів, як у випадку Anthropic). Це забезпечує відповідність інструкціям користувача й зменшує генерацію шкідливого або нерелевантного контенту, що підвищує їхню надійність як інструментів для письменників;

– всі три LLM є універсальними генеративними моделями, здатними виконувати широкий спектр завдань: від генерації коду й сумаризації до ведення діалогу та, що є ключовим для дослідження, творчого письма.

Водночас, саме унікальні відмінності тренувальних даних, архітектурних оптимізацій і фокусу міркування (Reasoning) створюють підґрунтя для емпіричного дослідження їхніх комплементарних властивостей та потенційної синергії в складних творчих завданнях [29–31].

Відмінності між GPT-4o, Claude 4 Opus і Gemini 2.5 Pro, що мають значення для завдань художнього аналізу та генерації, узагальнено в таблиці 2.1.

Таблиця 2.1 – Огляд відмінностей між GPT-4o, Claude 4 Opus і Gemini 2.5 Pro

Критерій	**GPT-4o**	**Claude 4 Opus**	**Gemini 2.5 Pro**
1	2	3	4
Основний фокус розробки	Універсальна продуктивність, швидкість і стилістична якість	Контекст, логічна послідовність і безпека	Глибока інтеграція даних, масштабування контексту, багатокрокове міркування
Контекстне вікно (tokens)	Високе (128 тис.), оптимізоване для високої швидкості	Дуже високе (200 тис.), з акцентом на стабільність довгого контексту	Екстремально високе (до 1 млн.) ідеально для аналізу цілих романів/світів
Сильні сторони у генерації	Вишуканий художній стиль, емоційна насиченість, висока швидкість відповіді. Ідеально для фінальної прози	Логічна послідовність, цілісність сюжетної лінії та характеру, глибокий аналіз мотивацій	Ефективна інтеграція великих обсягів вхідних даних, детальний планувальник сцени

Продовження таблиці 2.1

1	2	3	4
Очікувані слабкі сторони	Ризик «загальності» чи «полірування» стилю, що може нівелювати гостроту персонажів	Можлива «обережність» у генерації через фільтри безпеки, що може обмежувати художню непередбачуваність	Необхідність перевірки стилістичної витонченості українською мовою у порівнянні з конкурентами
Контроль температури (determinism)	Середній. Схильність до творчості, що іноді ускладнює повний контроль	Високий. Тренування на безпеці сприяє більш передбачуваній і контрольованій поведінці	Середній/Високий. Здатність до логічного планування забезпечує певний рівень передбачуваності
Доступність та технічні вимоги (API/вартість)	Висока доступність через API. Оптимізована швидкість може знижувати вартість токенів	Висока доступність через API. Вища вартість через акцент на довгому контексті й міркуванні	Висока доступність через API. Конкурентна ціна, особливо для Flash-версій

Виходячи з проведеного аналізу спільного і відмінного між моделями, можна побачити, що всі три моделі мають свої переваги і недоліки. Задля виявлення найкращої моделі для аналізу й генерації художніх текстів буде про вироблено методику оцінки кожної моделі.

Крім того, буде запропоновано методику для покращення результатів аналізу та генерації, а конкретніше, спосіб гібридного використання сильних сторін моделей.

2.5 Формування методик для аналізу та генерації художніх текстів за допомогою великих мовних моделей

2.5.1 Формування набору вхідних даних для тестування великих мовних моделей

Для об’єктивного порівняльного дослідження обраних великих мовних моделей у контексті аналізу та генерації художніх текстів, треба створити набір вхідних даних. Цей набір має імітувати реальні умови, коли користувач (письменник) надає неструктуровані й розрізнені описи світу, персонажів і сюжету. Якість аналізу і генерації художнього тексту моделями прямо залежить від їх здатності ефективно вилучати ключові знання (knowledge retrieval) та коректно інтегрувати їх у фінальну сцену [32, 33].

Формування набору вхідних даних включає два основних компоненти: неструктуровані контексти й структурований користувацький запит (промпт).

2.5.1.1 Формалізація вхідних контекстів

Вхідні контексти моделюються як множина неструктурованих описів D, які містять усю необхідну інформацію про художній світ і його елементи. Неструктурований характер цих описів обраний для емпіричної перевірки ефективності механізмів RAG та Reasoning у тестованих LLM [34, 35].

Нехай множина неструктурованих описів D визначається як об’єднання підмножин, що описують ключові аспекти сюжету й світу:

$$D = \{D_{char}, D_{loc}, D_{rule}, D_{story}\}, \tag{2.1}$$

де D_{char} – множина описів персонажів, що включає мотивації, біографічні дані та психологічні характеристики:

$$D_{char} = \{d_1^{char}, \dots, d_{n_c}^{char}\}; \quad (2.2)$$

D_{loc} – множина описів локацій, яка містить їхні фізичні, архітектурні чи атмосферні особливості:

$$D_{loc} = \{d_1^{loc}, \dots, d_{n_l}^{loc}\}; \quad (2.3)$$

D_{rule} – множина описів правил, по яким працює світ в історії (worldbuilding), включаючи закони фізики, магічні системи, соціальні та політичні правила, закони, релігії, культури, технології чи сюжетні обмеження:

$$D_{rule} = \{d_1^{rule}, \dots, d_{n_r}^{rule}\}; \quad (2.4)$$

D_{story} – узагальнений опис того, що відбулося у сюжеті до даної сцени, яку планує писати користувач. Включає ключові події, що відбувалися до створюваної сцени (береться до уваги, щоб будувати нову сцену логічно узгоджено, підтримуючи послідовність і внутрішню узгодженість сюжету) [36].

Усі елементи d_i є текстовими фрагментами, котрі подаються моделі як єдиний, неупорядкований вхідний текст.

2.5.1.2 Структура користувацького запиту (пропмту)

Користувацький запит q містить цільову інструкцію для генерації сцени та керуючі параметри. Його компоненти спрямовані на перевірку здатності моделі до виконання складних багатоаспектних вимог.

Запит q може мати такі структурні компоненти:

$$q = \{g, s, C_{must}, C_{avoid}, P\}, \quad (2.5)$$

де g – мета генерації, що визначає сцену, яка має статися;

s – бажаний тон, жанр;

C_{must} – обов'язкові умови та/або події, що мають статися в сцені;

C_{avoid} – заборони та/або обмеження щодо сцени;

P – параметри генерації, до прикладу, бажана довжина згенерованого тексту або коефіцієнт температури (параметр, котрий контролює випадковість вибору наступного токена при генерації. Високий – вища ймовірність вибрати менш вірогідного чи креативного, але потенційно менш логічного чи послідовного слова. Низький – вибирається найвірогідніше слово, призводить до більш детермінованого, послідовного та логічного виводу).

2.5.2 Тестування великих мовних моделей на задачах аналізу і генерації художнього тексту

Розробляється спосіб отримання результатів аналізу та генерації від великих мовних моделей для подальшого їх використання у виявлення найкращої LLM для кожної з задач.

Методика ґрунтується на підході Zero-shot Learning з використанням загального вхідного промпту, що містить контекст D і запит користувача q для мінімізації впливу додаткового налаштування. Моделі будуть робити аналіз і генерувати художній текст, спираючись виключно на свої узагальнені знання й промпт, не отримуючи жодних конкретних прикладів бажаного стилю чи формату виводу в промпті, що дозволить об'єктивно оцінити їхню внутрішню здатність до виконання поставлених задач [37].

Спочатку формується вхідний запит. Спеціальний системний промпт об'єднується з контекстом D і користувацьким запитом q в єдиний великий промпт, що є ідентичним для всіх трьох моделей. Він подається моделі зі спеціалізованим системним промптом, який вимагає лише аналітичного виводу.

Системний промпт для аналізу формується таким чином:

«Ти – аналітична модель, завдання якої: з будь-якого наданого тексту витягти і структурувати всі ключові факти. Дотримуйся правил виводу. Структура документа:

– кожна основна категорія починається із заголовка «Сутність: Назва» (крім категорії «Можливі дії: Ім’я персонажа»). Персонажі: «Персонаж: Ім’я» Локації: «Локація: Назва». Країни: «Країна: Назва». Артефакти: «Артефакт: Назва». Магічні системи: «Магія: Назва». Будь-які інші категорії – за потреби;

– якщо у тексті декілька персонажів або локацій, кожна сутність отримує окремий блок;

– усередині кожної категорії кожний факт – окремий пункт «Ключ: Значення».

Біографії та тимчасові події мають розділятися на атомарні факти. Емоції, характер і психологічні аспекти – кожен аспект записується окремим фактом, без довгих абзаців. Можливі дії персонажів:

– після виводу всіх фактів для всіх сутностей додається окремий розділ для можливих дій персонажів;

– виділяються можливі дії персонажів (відповідні їх характеру, роду діяльності, передісторії, правилам світу) у сцені із запиту, наданого користувачем. Не потрібно писати саму сцену, лише можливі дії;

– для кожного персонажа створюється окремий блок («Можливі дії: Ім’я персонажа», кожний пункт дії позначається через «-», короткий опис можливої дії персонажа в сцені.

Не додається жоден вільний текст чи пояснення, не слід намагатися підсумовувати або інтерпретувати, тільки факти.».

Системний промпт для генерації формується таким чином:

«Ти – літературний автор. Твоя мета – створювати художні сцени високої якості за запитом користувача.

Дотримуйся наступних вимог:

− текст повинен бути цілісним і логічним;

− використовуй художній, образний стиль, а не технічний опис;

– тон тексту повинен відповідати зазначеному в запиті користувача;

– діалоги персонажів мають бути живими, правдоподібними і розкривати характери;

– не виходь за межі наданого запиту користувача. Якщо у запиті бракує деталей, то додай їх самостійно, але так, щоб це виглядало природно;

– завжди завершуй сцену логічною крапкою – не залишай її обірваною.

Напиши художню сцену, яка відповідає запиту користувача.».

Тестування проводиться для кожної множини обраних моделей

$$\{LMM_1, LLM_2, LLM_3\}, \tag{2.6}$$

де LMM_1 – модель GPT-4o;

LMM_2 – модель Claude 4 Opus;

LMM_3 – модель Gemini 2.5 Pro.

Далі відбувається отримання відповіді від моделі: в залежності від тесту, який наявний, модель отримує відповідний сформований промпт і надає результат аналізу або генерації:

$$O_A = LLM(Sp_A, D, q), \tag{2.7}$$

де O_A – результат аналізу;

Sp_A – системний промпт для аналізу.

$$O_G = LLM(Sp_G, D, q), \tag{2.8}$$

де O_G – результат аналізу;

Sp_G – системний промпт для генерації.

Результат зберігається для подальшого використання в оцінці якості.

2.5.3 Метод оцінки якості аналізу контексту

Методика оцінки якості аналізу контексту спрямована на об’єктивне вимірювання здатності великих мовних моделей ефективно вилучати, структурувати та логічно узгоджувати ключові знання з наданого промпту.

Отримана оцінка дасть зрозуміти, чи є місце для покращення існуючого способу отримання аналізу тексту від моделей. Якщо так, вона послужить для виявлення кращої моделі-аналітика, яка може забезпечувати достовірність психології персонажів і логіки світу в подальшій пропонованій системі комбінування роботи моделей.

Оцінка здійснюється за допомогою автоматизованого інструментарію шляхом порівняння структурованого виводу-списку моделі O_A з еталонним аналізом – списком, створеним експертом, що містить ідеально вилучені факти та логічні зв’язки.

Якість аналізу вимірюється як комплексний показник, що поєднує шість науково визнаних метрик, які охоплюють повноту, надійність і релевантність вилученої інформації.

Будуть використовуватися такі метрики:

– повнота покриття контексту (Coverage Score). Ця метрика є індикатором ефективності механізму RAG моделі та її здатності використовувати довге контекстне вікно. Вона розраховується за формулою

$$Coverage = \frac{Facts_{extracted} \cap Facts_{ref}}{Facts_{ref}}, \quad (2.9)$$

де $Facts_{extracted}$ – множина вилучених фактів;

$Facts_{ref}$ – множина еталонних фактів.

Метрика оцінює, наскільки повно модель вилучила всі ключові факти та правила з вхідному контексті D. Високий показник мінімізує ризик «забування» LLM критично важливих деталей, які мають впливати на сюжет [38];

– рівень галюцинацій (Hallucination Rate). Метрика визначає частку інформації у виводі, яка не має прямого підтвердження у вхідному контексті D. Це ключовий показник надійності аналітичного апарату моделі. Розраховується як

$$Hallucination\ Rate = \frac{neutral + contradiction}{total}, \tag{2.10}$$

де $neutral$ – кількість нейтральних тверджень;

$contradiction$ – кількість суперечливих тверджень;

$total$ – загальна кількість тверджень.

Моделі з низьким Hallucination Rate демонструють вищу точність і надійність, що є критичним для запобігання сюжетним помилкам (Lore Breaking) [39];

– точності виділення сутностей (Entity Linking Accuracy). Метрика оцінює коректність ідентифікації та класифікації унікальних сутностей (персонажі, локації, артефакти, правила).

Вимірюється точність (Precision), повнота (Recall) і $F1$-міра на рівні сутностей. Для їх розрахунку використовуються такі показники як True Positives (TP) – кількість сутностей, які коректно виділені моделлю (знайдені моделлю і збігаються з еталоном), False Positives (FP – кількість сутностей, які некоректно виділені моделлю (знайдені моделлю, але відсутні еталоні, бо модель помилково позначила слово як сутність або дала неправильний тип), False Negatives (FN): кількість сутностей, які пропущені моделлю, але є в еталоні.

Точність показує, наскільки достовірні виділені моделлю сутності:

$$Precision = \frac{TP}{TP + FP}. \tag{2.11}$$

Повнота показує, наскільки повно модель виділила всі сутності з еталону:

$$Recall = \frac{TP}{TP + FN}. \quad (2.12)$$

$F1$-міра – усереднений показник, який є головною метрикою якості виділення сутностей (гармонійне середнє):

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}. \quad (2.13)$$

Також експертами буде оцінюватися логічність виділення моделями можливих дій для кожного персонажа в контексті сцени із запиту q. Експерти будуть давати свою оцінку за Шкалою Лікерта – психометричною шкалою, що використовується для вимірювання ставлень чи думок, де респонденти оцінюють ступінь своєї згоди чи незгоди із твердженням, обираючи один із фіксованих пунктів (від 1 – «дуже погано» до 5 – «винятково добре»). Середнє значення оцінок експертів $Actions\ Logic\ Score$ буде враховуватися в інтегральній оцінці.

Інтегральний показник якості аналізу (Q_A) для кожної моделі обчислюється як зважений комплексний індекс на основі шести метрик, що дозволяє визначити модель-аналітика. Визначається як

$$\begin{aligned} Q_A = {} & \omega_1 \cdot Coverage + \omega_2 \cdot (1 - Hallucination\ Rate) + \\ & + \omega_3 \cdot Consistency\ Score + \omega_4 \cdot F1 + \omega_5 \cdot Actions\ Logic, \end{aligned} \quad (2.14)$$

де ω_n – ваговий коефіцієнт, що визначається експертно та відображає наскільки важлива і чи інша якість для виявлення кращої моделі.

2.5.4 Метод оцінки якості художньої генерації

Мета проведення оцінки якості художньої генерації – емпірично визначити чи здатні обрані великі мовні моделі на написання складних, якісних, цікавих літературних творів.

Окрім того, ця оцінка покликана знайти кращу модель для цієї задачі – модель-генератор, що демонструє найвищу якість прози, стилістичну вишуканість та емоційну насиченість, і буде використана в подальшому методі комбінації LLM.

Оскільки художня якість є за своєю суттю суб'єктивною категорією, основний фокус методики зміщується на детальну експертну оцінку (Human Evaluation) [40].

Традиційні автоматичні метрики (BLEU, ROUGE) ефективні для оцінки схожості тексту з еталоном, але нездатні адекватно виміряти ключові художні параметри, такі як:

- оригінальність і креативність;
- емоційний вплив та атмосфера;
- природність та правдоподібність діалогів.

Тому для оцінки виводу O_G застосовується багатофакторний метод експертного оцінювання.

2.5.4.1 Формування критеріїв експертної оцінки

Експерти оцінюють кожну згенеровану сцену за чотирма основними критеріями за Шкалою Лікерта.

Оцінюватися будуть такі критерії:

- літературна вишуканість (Prose Quality, P_1) – складність та образність мови, різноманітність синтаксису, відсутність тавтологій та стилістичних помилок;

– емоційна насиченість (Emotional Depth, P_2) – здатність тексту викликати емоційний відгук, глибина розкриття почуттів персонажів, відповідність тону, заданому в запиті q;

– психологічна достовірність (Plausibility, P_3) – правдоподібність дій та діалогів у контексті сцени. Оцінка того, наскільки дії персонажів логічно узгоджені з їхніми мотиваціями і правилами світу;

– стилістична імітація (Tone Adherence, P_4) – наскільки згенерований текст відповідає жанру та стилістичним вимогам s, вказаним у запиті користувача q;

– технічна коректність та грамотність (Technical Literacy, P_5) – відсутність орфографічних, пунктуаційних, граматичних і синтаксичних помилок. Фактична узгодженість у межах самого тексту (наприклад, чи не змінюється колір очей персонажа) [41].

2.5.4.2 Методика проведення експертної оцінки

Для кожної моделі LLM_i обчислюється фінальний показник якості генерації Q_G. Для кожного критерію P_j, де $j = 1, \dots 5$, та кожного O_G. Загальний показник Q_G розраховується як зважене арифметичне середнє п’яти критеріїв:

$$Q_G(LLM_i) = \sum_{j=1}^{5} \omega_j \cdot \overline{P_j}\,, \qquad (2.15)$$

де ω_j – ваговий коефіцієнт, експертно визначений ваговий коефіцієнт;

$\overline{P_j}$ – усереднена оцінка за критерієм.

2.5.5 Формування та обґрунтування методу комбінованого використання великих мовних моделей

Якщо буде виявлено, що жодна з моделей не обходить інші в обох категоріях аналізу і генерації, пропонується метод комбінованого використання великих мовних моделей. Він розроблений для подолання обмежень окремих моделей у виконанні складних, багатоаспектних завдань, що вимагають як глибокого логічного аналізу, так і високоякісної художньої генерації. Метод базується на архітектурі, що використовує модель-аналітика та модель-генератор, які визначаються тестами і методами оцінки.

Дана методика забезпечує перетворення неструктурованих вхідних даних (D, q) на фінальний художній текст T через послідовність проміжних, структурованих форм, із застосуванням ітераційної валідації. Процес поділяється на три ключові етапи: аналіз, генерація та валідація/ітерація [42].

Етап 1. Формалізований аналіз. Модель-аналітик виконує логіко-семантичний аналіз кожного елемента світу й будує надійну аналітичну модель сцени. Для кожного персонажа c_i модель виконує логіко-семантичний аналіз його опису d_i^{char} у контексті всієї інформації:

$$Descr_i = LLM_A\left(d_i^{char}, q, D_{loc}, D_{rule}, D_{story}\right), \quad i = 1..n, \tag{2.16}$$

де LLM_A – модель-аналітик;

$Descr_i$ – структурована інтерпретація, що агрегує критичні елементи для створення достовірного профайлу персонажа:

$$Descr_i = \{M_i, A, C_i, R_i, L_i, B_i\}, \tag{2.17}$$

де M_i – множина мотивацій персонажа;

A – опис зовнішності;

C_i – множина рис характеру;

R_i – роль/статус у світі;

L_i – лексичні й стилістичні маркери (типова лексика, фразеологія, синтаксис);

B_i – минуле персонажа (backstory).

LLM_A використовує структурований опис $Descr_i$ та обмеження з промпту q для прогнозування логічно узгодженої множини допустимих дій кожного персонажа в сцені:

$$Act_i = LLM_A\big(Descr_i, q, D_{loc}, D_{rule}, D_{story}\big), \tag{2.18}$$

при чому в Act_i беруться обов'язкові дії персонажа C_{must} і заборони C_{avoid} із промпту q. Результати аналізу для всіх персонажів агрегуються в єдину структуровану модель сцени:

$$Scene = \big\{Char_1, \dots Char_n, q, D_{loc}, D_{rule}, D_{story}\big\}, \tag{2.19}$$

де $Char_i = \{Descr_i, Act_i\}$ – загальний опис персонажа.

Модель-аналітик формує фінальний промпт $Prompt$ для моделі-генератора, використовуючи модель Scene як основний структурований контекст:

$$Prompt = LLM_A(Scene, I), \tag{2.20}$$

де I – додаткові технічні інструкції для контролю генерації.

Етап 2. Створення художнього тексту. Модель-генератор LLM_G отримує і враховує всі параметри сцени $Scene$ і створює текст T:

$$T = LLM_G(Prompt). \tag{2.21}$$

Етап 3. Валідація й ітеративна корекція (Quality Control). Цей етап гарантує, що кінцевий текст T відповідає високим вимогам узгодженості та художньої якості, застосовуючи цикл валідації та ітерації. Валідація проводиться за тими самими метриками, що використовуються при оцінці якості генерації художнього тексту. Якщо оцінка якості не досягає заданого порога, текст повертається на повторну генерацію, де модель використовує попередній невдалий вивід T як додатковий контекст для корекції.

Схему методу комбінованого використання великих мовних моделей зображено на рисунку 2.1.

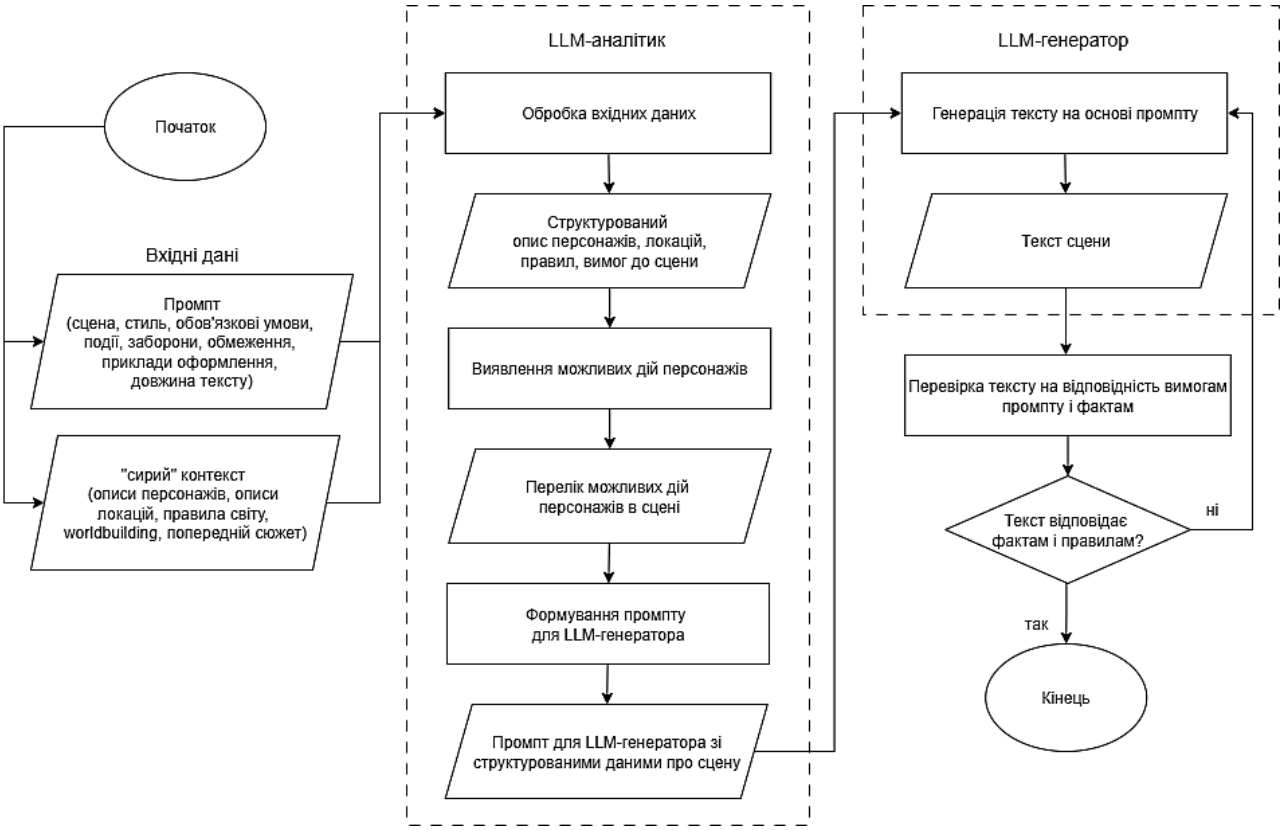

Рисунок 2.1 – Схема методу комбінованого використання великих мовних моделей для аналізу та генерації художніх текстів

Запропонований метод комбінованого використання великих мовних моделей базується на факті, що ефективність LLM диференційована за типом завдання: одна модель може бути кращою в глибокому логічному аналізі, а інша – у художній вишуканості.

Методика усуває недоліки монолітного підходу шляхом розділення обов’язків між ідентифікованими моделлю-аналітиком і моделлю-генератором. Аналітик створює верифікований, структурований план, який мінімізує ризик галюцинацій і сюжетних суперечностей.

Генератор використовує цей надійний план як високоякісний контекст, концентруючись виключно на стилістичній якості та емоційній глибині тексту. Таке послідовне двофазне оброблення максимізує сильні сторони кожної моделі, і забезпечує необхідну логічну узгодженість та контроль якості художнього виводу [43].

2.6 Моделювання структури програмного застосунку для аналізу та генерації художніх текстів за допомогою великих мовних моделей

2.6.1 База даних

Для ефективного проведення експериментів з великими мовними моделями й оцінки їхніх відповідей розробляється структура бази даних. Ця БД забезпечує зберігання вхідних тестових сценаріїв, даних про використані LLM, кількісні й експертні результати, отримані на етапах аналізу, генерації та гібридного методу.

Обрано модель реляційної БД, оскільки вона гарантує цілісність даних, забезпечує чітку структуру зв’язків між об’єктами (тестовими сценаріями й результатами) та спрощує подальшу агрегацію й аналіз експериментальних метрик.

База даних складається з п’яти таблиць, зв’язок між якими реалізовано за допомогою зовнішніх ключів.

Таблиця LLM_Models необхідна для збереження усіх великих мовних моделей, що беруть участь в експерименті. Структура таблиці описана в таблиці 2.2.

Таблиця 2.2 – Структура таблиці LLM_Models

Поле	Тип даних	Опис
model_id	INTEGER	Первинний ключ – унікальний ідентифікатор моделі.
model	TEXT	Назва моделі.
model_name	TEXT	Технічний ідентифікатор, котрий використовується в API-запитах до моделей.

Таблиця Test_Scenarios містить повний набір вхідних даних, що використовуються для тестування. Кожний запис є унікальним експериментальним сценарієм, що використовується для оцінки відповідей моделей. Структура таблиці описана в таблиці 2.3.

Таблиця 2.3 – Структура таблиці Test_Scenarios

Поле	Тип даних	Опис
scenario_id	INTEGER	Первинний ключ – унікальний ідентифікатор тестового сценарію.
context_d	TEXT	Вхідний текст, що підлягає аналізу.
prompt_q	TEXT	Текстовий запит від користувача до моделі.
gold_ref	TEXT	Еталонний вивід, створений експертом.
test_type	TEXT	Тип сценарію: «analysis» (аналіз), «generation» (генерація) або «hybrid» (комбінування моделей).

Таблиця Analysis_Results зберігає результати оцінки моделей на етапі аналізу художнього тексту. Ці дані необхідні для оцінки здібностей моделей до аналізу тексту і для вибору оптимальної моделі-аналітика в методі комбінування моделей. Структура таблиці описана в таблиці 2.4.

Таблиця Generation_Results призначена для зберігання результатів тестування моделей на етапі генерації художнього тексту, що використовуються для оцінки здібностей моделей до генерації якісного художнього тексту і для вибору моделі-генератора в методі комбінування моделей. Структура таблиці описана в таблиці 2.5.

Таблиця 2.4 – Структура таблиці Analysis_Results

Поле	**Тип даних**	**Опис**
analysis_result_id	INTEGER	Первинний ключ.
scenario_id	INTEGER	Зовнішній ключ до Test_Scenarios.
model_id	INTEGER	Зовнішній ключ до LLM_Models.
output_o_an	TEXT	Згенерований LLM аналіз.
coverage_score	REAL	Показник метрики Coverage.
hallucination_rate	REAL	Показник метрики Hallucination Rate.
entity_f1_score	REAL	Показник метрики F1.
actions_logic_score	REAL	Показник метрики Actions Logic.
q_an_final	REAL	Фінальна оцінка якості аналізу.

Таблиця 2.5 – Структура таблиці Generation_Results

Поле	**Тип даних**	**Опис**
generation_result_id	INTEGER	Первинний ключ.
scenario_id	INTEGER	Зовнішній ключ до Test_Scenarios.
model_id	INTEGER	Зовнішній ключ до LLM_Models.
output_text	TEXT	Згенерований художній текст (продовження історії).
p1	REAL	Показник метрики Prose Quality.
p2	REAL	Показник метрики Emotional Depth.
p3	REAL	Показник метрики Plausibility.
p4	REAL	Показник метрики Tone Adherence.
p5	REAL	Показник метрики Technical Literacy.
q_gen_final	REAL	Фінальна комбінована оцінка якості генерації.

Таблиця Hybrid_Results фіксує результати, отримані за допомогою розробленого гібридного підходу, який поєднує можливості моделі-аналітика та моделі-генератора. Структура таблиці описана в таблиці 2.6.

Таблиця 2.6 – Структура таблиці Hybrid_Results

Поле	**Тип даних**	**Опис**
hybrid_result_id	INTEGER	Первинний ключ.
scenario_id	INTEGER	Зовнішній ключ до Hybrid_Scenarios.
output_an_hybr	TEXT	Згенерований моделлю-аналітиком аналіз.
output_text_hybr	TEXT	Фінальний текст, згенерований гібридним методом.
p1	REAL	Показник метрики Prose Quality.
p2	REAL	Показник метрики Emotional Depth.
p3	REAL	Показник метрики Plausibility.
p4	REAL	Показник метрики Tone Adherence.
p5	REAL	Показник метрики Technical Literacy.
q_gen_final	REAL	Фінальна комбінована оцінка якості генерації.

Структура бази даних зображена на рисунку 2.2.

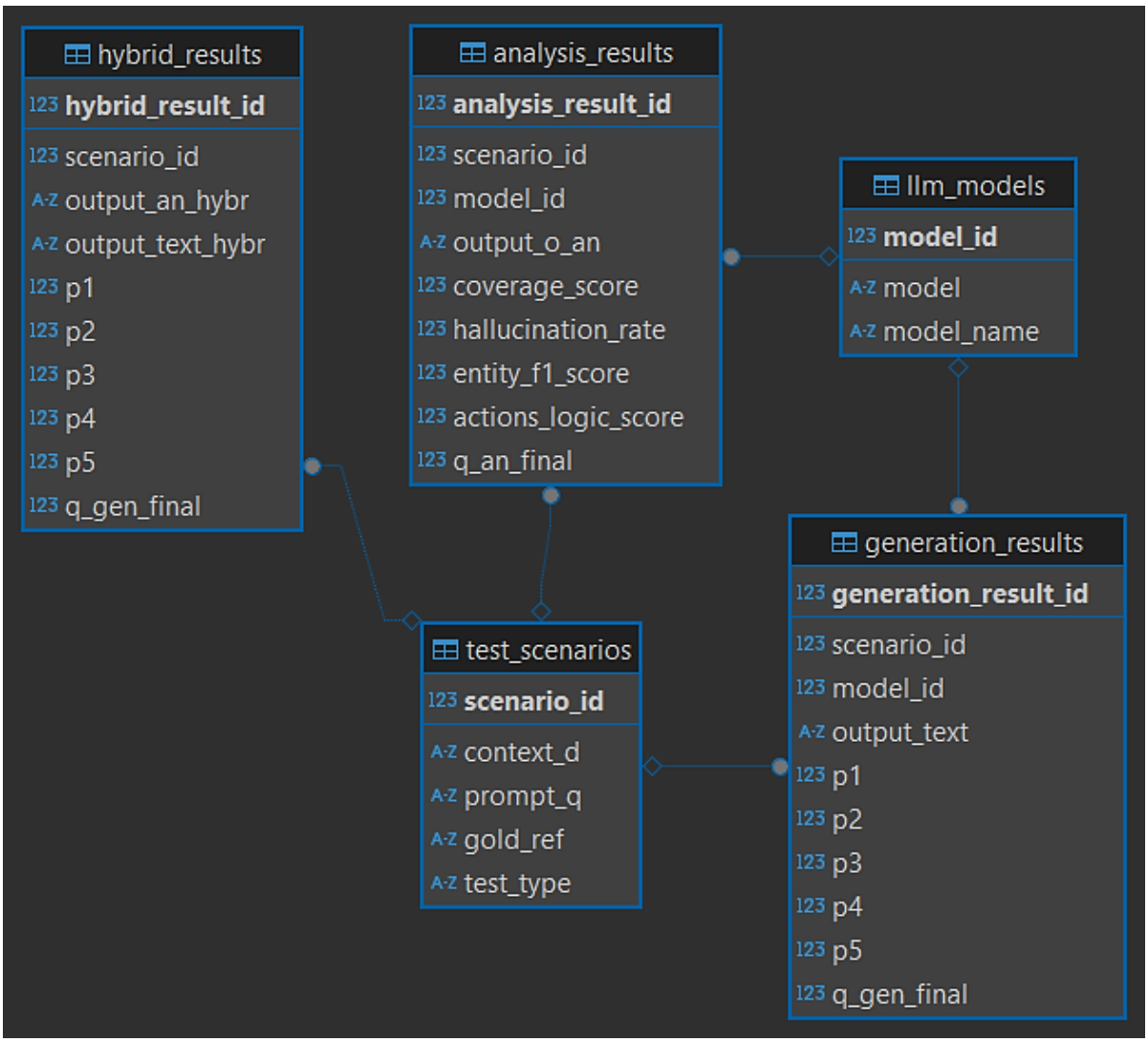

Рисунок 2.2 – Схема таблиць бази даних

2.6.2 Структура застосунку

Архітектура застосунку побудована на модульному принципі, що забезпечує гнучкість конфігурації, розширення функціоналу оцінювання та легкість інтеграції нових моделей.

Ключові функціональні частини застосунку включають:

– модуль конфігурації й ініціалізації, що відповідає за встановлення зовнішніх зв’язків і визначення глобальних параметрів середовища. Тут відбувається конфігурація API й БД – встановлення параметрів, таких як ключі доступу до API та шлях до файлу бази даних;

– модуль управління даними й зберігання, що гарантує цілісність, нормалізацію та систематизацію експериментальних даних. У ньому відбувається ініціалізація й підтримка бази даних, обслуговування запитів – надання інтерфейсу для операцій запису нових експериментальних сценаріїв, фіксації отриманих метрик і вибіркового отримання даних для подальшого аналізу;

– модуль взаємодії з моделями, котрий керує безпосереднім спілкуванням з моделями, забезпечуючи подачу інформації до них;

– модуль оцінювання та метрик, котрий забезпечує вимірювання якості виводу моделей (проводить оцінку по метриках для оцінки якості аналізу і генерації тексту моделями);

– модуль координації тестів і звітності, який координує виконання всього експериментального робочого процесу. Має функціонал для ініціації та керування як одиночними тестовими прогонами (для конкретного сценарію та моделі), так і повним циклом тестування, що послідовно охоплює всі вхідні сценарії. Має функціонал для вилучення, агрегації й подання результатів з бази даних у формі, зручній для порівняльного порівняння результатів тестування моделей.

3 ДОСЛІДЖЕННЯ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ ДЛЯ АНАЛІЗУ ТА ГЕНЕРАЦІЇ ХУДОЖНІХ ТЕКСТІВ

3.1 Вибір інструментальних засобів для реалізації поставлених задач

Для реалізації експериментальної частини дослідження обрано середовище Google Colab, що забезпечує зручну інтеграцію з мовою програмування Python і дозволяє працювати з сучасними бібліотеками машинного навчання без необхідності налаштування локальної інфраструктури. Використання Colab спрощує процес тестування великих мовних моделей, оскільки надає доступ до графічних процесорів (GPU), підтримує асинхронні виклики до API моделей й інтегрується з хмарними сховищами для збереження результатів експериментів.

Основною мовою реалізації є Python 3.10. Цей вибір обґрунтований його домінуючою позицією в сфері наукових обчислень, зокрема, у галузі машинного навчання і штучного інтелекту.

Для роботи з моделями використано бібліотеку PyTorch, бо вона забезпечує високу продуктивність обчислень на GPU та дозволяє ефективно працювати з тензорними структурами даних. Ця бібліотека надає інструменти для завантаження попередньо натренованих моделей і подальшої адаптації їх до конкретних завдань дослідження. Зокрема, на основі PyTorch інтегровано пакети Transformers та Sentence-Transformers.

Бібліотека Transformers від компанії Hugging Face використовувалася для роботи з моделями класифікації послідовностей, токенізації вхідних текстів і обчислення семантичної подібності між фрагментами. Вона забезпечує уніфікований інтерфейс для багатьох моделей, що спрощує експериментування і порівняння різних архітектур.

Модуль Sentence-Transformers застосовувався для побудови багатомовних векторних подань речень, зокрема, моделей distiluse-base-multilingual-cased-v2 та paraphrase-multilingual-MiniLM-L12-v2.

Обидві моделі підтримують українську, англійську та інші європейські мови, що дозволяє досліджувати якість LLM при роботі з багатомовними корпусами художніх текстів.

Модель distiluse-base-multilingual-cased-v2 є спрощеною версією мультимовної USE (Universal Sentence Encoder), оптимізованої для швидкого отримання якісних семантичних векторів. Вона підтримує понад 50 мов, включно з українською, англійською, німецькою, французькою та іншими європейськими мовами.

Модель paraphrase-multilingual-MiniLM-L12-v2, базується на архітектурі MiniLM, що забезпечує вищу продуктивність при менших обчислювальних витратах. Вона створена для задач пошуку парафраз, класифікації текстів і кластеризації, а також демонструє стабільні результати для коротких і довгих фрагментів художнього тексту.

Використання цих двох моделей дозволяє порівняти якість багатомовних ембедингів і оцінити здатність великих мовних моделей зберігати семантичну цілісність змісту при перекладі чи аналізі текстів різними мовами.

Для збереження результатів експериментів обрано вбудовану реляційну систему керування базами даних SQLite. Вона забезпечує простоту використання, автономність і відсутність потреби у серверному налаштуванні. Вона зберігає структури таблиць із результатами аналізу, сценаріями, оцінками моделей і дозволяє виконувати SQL-запити безпосередньо в середовищі Colab.

Асинхронна взаємодія з зовнішніми мовними моделями реалізована через клієнт AsyncOpenAI, який використовується для доступу до API платформи OpenRouter. Це надає можливість викликати обрані великі мовні моделі (зокрема, GPT-4o, Claude 4 Opus, Gemini 2.5 Pro) у єдиному інтерфейсі, що полегшує порівняння їх ефективності на однакових сценаріях.

Для статистичного аналізу результатів і побудови таблиць використано бібліотеку Pandas, яка надає зручні інструменти роботи з табличними структурами, фільтрації даних [44].

Обрані інструменти дозволять створити гнучке середовище для тестування оцінки результатів виводу великих мовних моделей у процесі аналізу й генерації художніх текстів.

3.2 Етапи програмної реалізації аналізу та генерації художніх текстів за допомогою великих мовних моделей

3.2.1 Архітектура системи

Розроблена система побудована за модульним принципом, що забезпечує її гнучкість, масштабованість та можливість поступового розвитку. Основною метою архітектури є створення середовища, де дані проходять багаторівневу трансформацію: від первинного сприйняття тексту до формування структурованого знання і генерації нового творчого матеріалу.

Архітектура передбачає розділення функцій на структурні блоки:

– блок роботи з базою даних і тестовими сценаріями. Реалізує створення структури бази даних, функції запитів до неї та формування тестових сценаріїв для експериментів. Забезпечує надійну організацію вхідних даних і контроль їх коректності;

– блок взаємодії з моделями, що відповідає за надсилання запитів до моделей через API й управління системними промптами. Забезпечує стандартизовану передачу інформації і правильну інтерпретацію запитів моделями;

– блок отримання результату аналізу й оцінки його якості. Виконує оцінку якості аналізу тексту за допомогою спеціальної метрики. Забезпечує отримання кількісних характеристик роботи моделей, що надалі використовуються для покращення генерації;

– блок отримання результату генерації й оцінки його якості, відповідає за створення текстових фрагментів на основі вхідних даних і отриманого аналізу. Паралельно здійснюється оцінка якості згенерованого тексту, що дозволяє контролювати точність та стилістичну відповідність результату;

– блок тестування методу комбінування великих мовних моделей, оцінки якості результатів його роботи. Він об’єднує попередні кроки у єдиний експериментальний цикл: надсилає запит на аналіз, передає отримані результати для генерації тексту та здійснює підсумкову оцінку якості. Цей блок дозволяє перевіряти ефективність комбінованого підходу до аналізу і генерації тексту;

– блок візуалізації результатів, що виводить отримані результати у вигляді графіків і таблиць.

Взаємодія блоків застосунку зображена на рисунку 3.1.

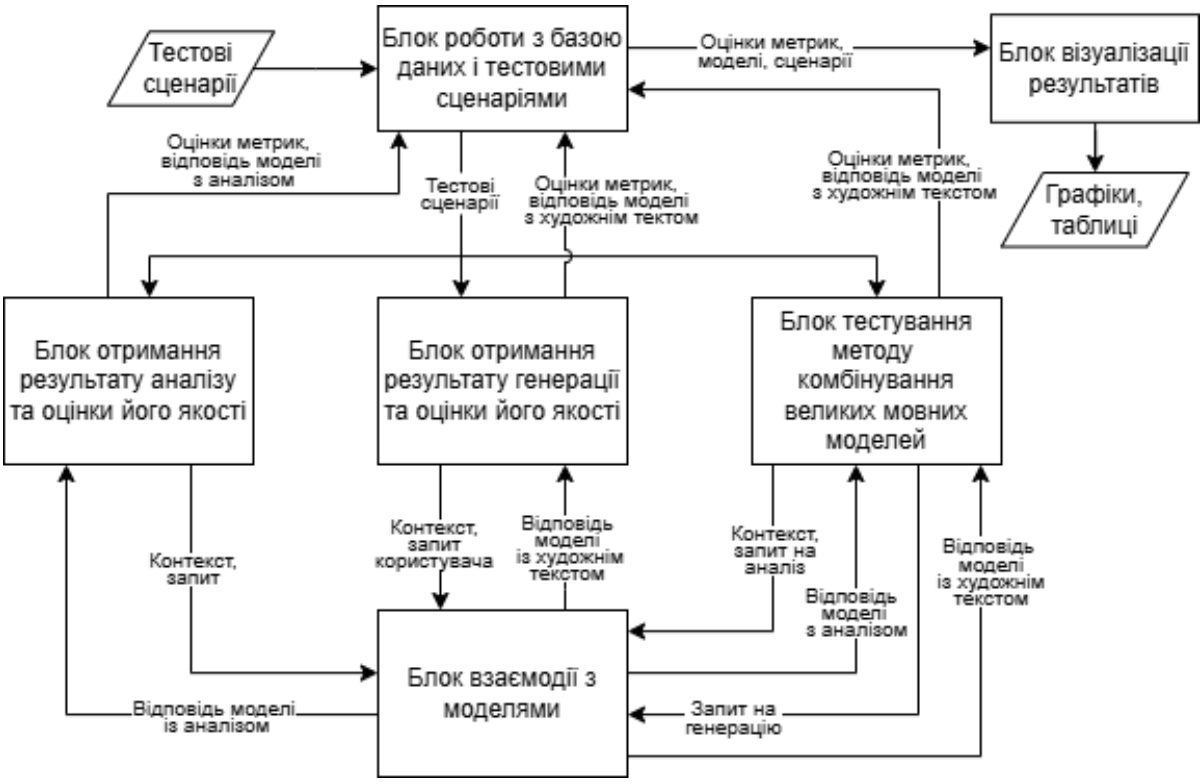

Рисунок 3.1 – Взаємодія блоків застосунку

Технічні рішення спрямовані на ефективне поєднання обчислювальної потужності та масштабованості. Система підтримує асинхронне виконання завдань, що дозволяє одночасно обробляти кілька потоків даних і запитів, знижуючи час очікування і підвищуючи продуктивність експериментів. Окрім того, архітектура забезпечує просту інтеграцію різних мовних моделей й інструментів, що дозволяє гнучко змінювати конфігурації та розширювати функціонал без необхідності істотних змін у структурі системи.

3.2.2 Блок роботи з базою даних і тестовими сценаріями

Блок роботи з базою даних і тестовими сценаріями є основою для керування даними проєкту та забезпечення відтворюваності результатів. Основним завданням цього блоку є керування тестовим набором (сценаріями) та зберігання всіх проміжних і кінцевих результатів експериментів.

Функція setup_database() створює з'єднання з БД та ініціалізує п'ять таблиць (LLM_Models, Test_Scenarios, Analysis_Results, Generation_Results, Hybrid_Results. Збереженів таблиці Test_Scenarios дані будуть використовуватись в подальшому в блоках аналізу, генерації й комбінованого методу. Дані з таблиць результатів використовуватимуться в блоці візуалізації.

Реалізовано функції для атомарного додавання даних: insert_model (додавання моделі до LLM_Models), insert_scenario (додавання нового тестового сценарію до Test_Scenarios) і спеціалізовані функції для запису метрик і виводів у відповідні таблиці результатів: insert_analysis_result, insert_generation_result і insert_hybrid_result. Це дозволяє зберігати результати експериментів незалежно від того, на якому етапі вони були отримані.

Блок включає функції для вибірки і агрегації інформації. Функція get_scenario_by_id забезпечує отримання повного тестового набору за унікальним ідентифікатором, що є необхідним для повторного запуску моделювання та оцінки. Загальна функція get_all_results використовується для зведення всіх даних для подальшого порівняльного аналізу.

В цьому блоці до бази даних додаються тестові сценарії.

3.2.3 Блок взаємодії з моделями

Блок взаємодії з моделями забезпечує зв'язок програми з великими мовними моделями, які використовуються в дослідженні. Для уніфікації доступу до різноманітних моделей від різних провайдерів було обрано сервіс-агрегатор OpenRouter.

Він дозволяє використовувати єдиний асинхронний клієнт для доступу до десятків відкритих та комерційних моделей, мінімізуючи накладні витрати на інтеграцію окремих API для кожної моделі. Це значно спрощує процес порівняльного тестування.

Основою блоку є асинхронна функція get_llm_response, що слугує універсальним шлюзом для всіх LLM-взаємодій. Ця функція приймає назву моделі (model_name), список повідомлень (messages), що включає системний та користувацький промпти, а також параметр temperature для контролю креативності виводу. Її реалізація наведена у лістингу 3.1.

Лістинг 3.1 Реалізація функції універсального запиту до великих мовних моделей get_llm_response:

```
async def get_llm_response(model_name: str, messages: List[Dict[str, Any]], temperature: float = 0.7) -> str:
    try:
        completion = await client.chat.completions.create(
            extra_headers=OPENROUTER_HEADERS,
            model=model_name,
            messages=messages,
            temperature=temperature,
            timeout=120
        )
        return completion.choices[0].message.content
    except Exception as e:
        return f"Помилка LLM: {e}"
```

Для виконання завдання аналізу і генерації розроблена спеціалізована функція get_llm_output. Її ключова роль полягає у форматуванні вхідних даних відповідно до вимог дослідження:

– системний промпт. У ньому використовується змінна system_prompt, котра містить суворі інструкції щодо структури виводу. Цей промпт налаштовує LLM як аналітичну модель, що витягує всі ключові факти і структурує їх. До завдання аналізу в якості змінної system_prompt подається текст зі змінної SYSTEM_PROMPT_ANALYSIS, до завдання генерації – зі змінної SYSTEM_PROMPT_GENERATION;

– користувацький промпт, що формується шляхом об'єднання контексту D та запиту користувача q. Це гарантує, що модель має і повний опис світу, і точну вимогу до аналізу.

Реалізація функції наведена в лістингу 3.2.

Лістинг 3.2 Реалізація функції get_llm_output:

```
async def get_llm _output(model_name: str, system_prompt: str, context_d: str, prompt_q: str, temp) -> str:
    user_prompt = f"""
    Контекст:
    {context_d}
    Запит:
    {prompt_q}
    """
    messages = [
      {"role": "system", "content": system_prompt },
      {"role": "user", "content": user_prompt}
    ]
    try:
      llm_output = await get_llm_response(model_name, messages, temperature=temp)
      return llm_output
    except Exception as e:
      return f"Помилка LLM для {model_name}: {e}"
```

Для аналітичних завдань використовується низьке значення параметра temperature (0.0), що забезпечує детермінований і логічний вивід, що мінімізує креативні галюцинації у структурі фактів.

Для генерації зазвичай встановлюється вищий рівень temperature (наприклад, 0,7), що стимулює креативність та різноманітність генерованих сцен.

3.2.4 Блок отримання результату аналізу та оцінки його якості

Блок отримання результату аналізу й оцінки його якості відповідає за обчислення ключових метрик, що дозволяють кількісно оцінити здатність моделі до точного й повного вилучення інформації з контексту. Цей блок є виконавчим ядром першого етапу дослідження, завданням якого є автоматичне проходження всіх тестових сценаріїв усіма обраними моделями. Він інтегрує функції запиту до API, обчислення метрик і збереження результатів у базі даних, забезпечуючи кількісну оцінку якості аналізу.

Для оцінки виводу моделі розроблено три автоматичні метрики, що базуються на семантичній схожості, та одна експертна метрика. Усі вони вимагають парсингу виводу моделі, що реалізується в функції parse_facts.

Метрика Coverage оцінює повноту витягнутих фактів (model_values) у порівнянні з еталоном (gold_values). Вона вимірює частку еталонних фактів, для яких у виводі моделі знайдено семантично схожий факт (з косинусною схожістю більшою чи рівною 0,7). Для порівняння використовується повний рядок твердження «Сутність: Значення», щоб врахувати контекст. Реалізацію функції get_coverage, що підраховує Coverage наведено у лістингу 3.3.

Лістинг 3.3 Реалізація функції get_coverage:

```
def get_coverage (gold_ref: str, model_output: str, threshold: float = 0.7):
    # [...] (парсинг та формування gold_values і model_values)
```

```
    # Кодування фактів
    gold_emb = model.encode(gold_values, convert_to_tensor=True)
    model_emb = model.encode(model_values, convert_to_tensor=True)
    sim_matrix = util.cos_sim(gold_emb, model_emb)
    matched = 0
    total = len(gold_values)

    for i in range(total):
       # Максимальна схожість для кожного еталонного факту
       best_sim = torch.max(sim_matrix[i]).item()
       if best_sim >= threshold:
          matched += 1
    return round(matched / total if total > 0 else 0, 3)
```

Метрика Hallucination Rate вимірює частку витягнутих фактів, які не можуть бути підтверджені вхідним контекстом D.

Факт вважається галюцинацією, якщо його семантична схожість з кожним реченням контексту є нижчою за поріг 0,85. Реалізацію наведено у лістингу 3.4.

Лістинг 3.4 Реалізація функції обчислення Hallucination Rate:

```
def get_hallucination_rate (D_text: str, model_output: str, threshold: float = 0.85):
    # [...] (парсинг фактів model_facts)
    # Кодування фактів моделі та контексту
    gold_emb = embed_model.encode(gold_facts, convert_to_tensor=True, normalize_embeddings=True)
    model_emb = embed_model.encode(model_facts, convert_to_tensor=True, normalize_embeddings=True)
```

```
    sim_matrix = util.cos_sim(model_emb, gold_emb) # (model_facts x gold_facts)
    hallucinated = 0
    for i in range(sim_matrix.size(0)):
        best_sim = float(torch.max(sim_matrix[i]).item())
        # [...] (обробка негативних тверджень)
        # Звичайний факт: якщо немає схожого в gold – це галюцинація
        if best_sim < threshold:
            hallucinated += 1
    rate = hallucinated / len(model_facts)
    return rate
```

Метрика Entity *F*1 Score оцінює точність ідентифікації та класифікації ключових сутностей (наприклад, Персонаж, Локація) за допомогою міри *F*1, що базується на семантичному збігу сутностей між еталоном і виводом моделі.

Спочатку функція extract_entities витягує всі рядки заголовків («Тип: Назва») з еталонного та згенерованого текстів.

Оскільки модель може використовувати синоніми або незначні варіації в назвах сутностей, для порівняння використовуються векторні подання цих рядків. Збіг фіксується, якщо семантична схожість сутності з LLM-виводу та найкращої сутності з еталону перевищує поріг 0,85.

На основі цих збігів обчислюються класичні показники True Positives, False Positives і False Negatives, що потім використовуються для розрахунку Precision, Recall і фінальної міри *F*1. Обчислення відбуваються у функції get_entity_linking_metrics. Реалізацію цієї функції наведено у лістингу 3.5.

Лістинг 3.5 Реалізація функції get_entity_linking_metrics:

```
def get_entity_linking_metrics(gold_text: str, model_text: str, threshold: float = 0.85):
    gold_entities = extract_entities(gold_text)
```

```
    model_entities = extract_entities(model_text)
    if not gold_entities and not model_entities:
        return {"F1": 0.0}
    gold_embs = model.encode(gold_entities, convert_to_tensor=True)
    model_embs = model.encode(model_entities, convert_to_tensor=True)
    cosine_scores = util.cos_sim(model_embs, gold_embs)
    matched_gold = set()
    matched_model = set()
    for i, model_ent in enumerate(model_entities):
        best_idx = cosine_scores[i].argmax().item()
        best_sim = cosine_scores[i][best_idx].item()
        if best_sim >= threshold:
            matched_model.add(i)
            matched_gold.add(best_idx)
    TP = len(matched_model)
    FP = len(model_entities) - TP # False Positives:
    FN = len(gold_entities) - len(matched_gold)
    precision = TP / (TP + FP) if (TP + FP) > 0 else 0.0
    recall = TP / (TP + FN) if (TP + FN) > 0 else 0.0
    f1 = (2 * precision * recall / (precision + recall)) if (precision + recall) > 0
else 0.0
    return {
        # [...] (повернення інших метрик)
        "F1": round(f1, 4),
    }
```

На відміну від автоматичних метрик, які перевіряють повноту та вірність фактів, показник метрики Actions Logic Score оцінює прогностичну якість аналізу LLM. Він вимірює, наскільки логічно та правдоподібно запропоновані моделлю можливі дії персонажів відповідають їхньому психологічному профілю, характеристикам і обмеженням світу, витягнутим у процесі аналізу. Оскільки це є суб’єктивною інтерпретацією, для цього використовується експертна оцінка.

Оцінка відбувається вручну кількома незалежними експертами за шкалою від 1 до 10. Функція get_actions_logic_score відповідає за збір цих оцінок та їх нормалізацію. Функція очікує введення оцінок від експертів через кому. Далі відбувається нормалізація – отримується середнє арифметичне значення всіх введених оцінок. Для забезпечення сумісності з іншими метриками, що знаходяться у діапазоні [0,1], отримане середнє значення ділиться на 10.

Функція run_single_analysis_test керує повним циклом: ініціація запиту, отримання виводу, обчислення всіх метрик і збереження результату. Функція run_all_analysis_tests організовує проходження усіма всіх тестових сценаріїв.

3.2.5 Блок отримання результату генерації та оцінки його якості

Блок оцінки якості генерації відповідає за оцінку художньої майстерності та стилістичної відповідності текстів, згенерованих великими мовними моделями.

У цьому блоці є власні функції, що відповідають за цикл проведення тестів для кожної моделі і кожного тестового сценарію. Щодо метрик, то реалізована функція get_all_expert_scores. Вона використовується для збору оцінок експертів по п’ятьох метриках (P_1 – Prose Quality, P_2 – Emotional Depth, P_3 – Plausibility, P_4 – Tone Adherence, P_5 – Technical Literacy). Вона приймає назву метрики, збирає оцінки від експертів, і повертає середнє значення в діапазоні [0,1, 1]. Реалізацію функції отримання оцінки для однієї метрики наведено в лістингу 3.6.

Лістинг 3.6 Реалізація функції отримання експертної оцінки для однієї метрики:

```
def get_expert_score(metric_name: str) -> float:
    while True:
```

```
        raw = input(f"Введите оценки для {metric_name} (через запятую, 1–10): ").strip()
        try:
            scores = [float(x) for x in raw.split(",") if x.strip()]
            # [...] (перевірка діапазону)
            avg = sum(scores) / len(scores)
            return avg / 10
        except ValueError as e:
            # [...] (обробка помилки)
            pass
```

Загальна якість генерації обчислюється функцією compute_q_gen_final. Вона поєднує п’ять експертних оцінок (P_1−P_5) у зважену суму.

3.2.6 Блок тестування методу комбінування великих мовних моделей, оцінки якості результатів його роботи

У даному блоці реалізується тестування методу комбінування великих мовних моделей, і здійснюється оцінка якості його роботи, а саме художнього тексту, який отримується цим методом.

Функція get_best_models знаходить найкращі модель-аналітика і модель-генератор. Вона знаходить модель з найвищим показником якості аналізу Q_A і показник якості генерації Q_G. Реалізація функції наведена в лістингу 3.7.

Лістинг 3.7 Реалізація функції get_best_models:

```
def get_best_models():
    conn = sqlite3.connect(DB_FILE)
    cursor = conn.cursor()
    cursor.execute("""
```

```
        SELECT model_id FROM Analysis_Results
        WHERE q_an_final = (SELECT MAX(q_an_final) FROM Analysis_Results)
    """)
    best_analysis_model_id = cursor.fetchone()[0]
    cursor.execute("""
        SELECT model_id FROM Generation_Results
        WHERE q_gen_final = (SELECT MAX(q_gen_final) FROM Generation_Results)
    """)
    best_generation_model_id = cursor.fetchone()[0]
    cursor.execute("SELECT model_name FROM LLM_Models WHERE model_id=?", (best_analysis_model_id,))
    model_a_name = cursor.fetchone()[0]
    cursor.execute("SELECT model_name FROM LLM_Models WHERE model_id=?", (best_generation_model_id,))
    model_g_name = cursor.fetchone()[0]
    print(f"Найкраща аналітична модель: {model_a_name}")
    print(f"Найкраща генераційна модель: {model_g_name}")
    return best_analysis_model_id, best_generation_model_id, model_a_name, model_g_name
```

Модель-аналітик виконує раціональне завдання: вона структуровано описує наявні дані, виокремлює ключові факти, логічні зв’язки, причинно-наслідкові відношення, мотивації персонажів, смислові блоки. Її мета – зрозуміти, а не вигадати. Структурований аналіз виступає своєрідною когнітивною картою майбутнього тексту. Модель-генератор, натомість, спирається на результати аналізу, перетворюючи суху логічну структуру в художньо насичений, послідовний і змістовно точний текст. Вона творить, але в межах уже окресленої системи координат, що істотно підвищує зв’язність і концептуальну цілісність результату.

У функції run_combined_experiment обрана модель-аналітик отримує аналітичний промпт і видає структурований аналіз. Модель-генератор створює художній текст на основі нього. Після цього відбувається оцінка якості виводу.

Функція run_all_combined_experiments запускає тестування методу для кожного сценарію.

3.3 Застосування великих мовних моделей до вибраної предметної області

3.3.1 Промпти

Для отримання результатів від великих мовних моделей було розроблено три системні промпти.

Вони були розроблені таким чином, щоб максимально ефективно використовувати властивості моделі: здатність до точної екстракції даних (аналіз) і здатність до креативного синтезу (генерація). Повні версії промптів наведено в додатку А.

Системний промпт для аналізу (SYSTEM_PROMPT_ANALYSIS) був розроблений для реалізації етапу аналізу – перетворення неструктурованого або напівструктурованого тексту контексту на суворо формалізований набір даних. В промпті вказується вимога використовувати жорсткі маркери («##» для заголовків сутностей, «* Ключ: Значення» для фактів) гарантувала, що вихідний документ завжди матиме однакову структуру, незалежно від вхідних даних. Це мінімізувало ризик «галюцинацій» та неформалізованого виводу і дозволяло використовувати автоматичні метрики оцінки.

Моделі отримували інструкцію розбивати складні описи (наприклад, біографію, характер) на окремі, невеликі факти. Це забезпечувало повноту вилучення даних. Важливою частиною завдання аналізу, окрім вилучення всіх фактів про персонажів модель мала використати ці ж факти (характер, мотивація, поточна ситуація), було завдання запропонувати перелік логічних та психологічно обґрунтованих дій, які персонаж міг би виконати у заданій сцені.

Це виділялося спеціальним блоком («@@ Можливі дії: Ім’я Персонажа») і слугувало безпосереднім «місточком» до етапу генерації, надаючи автору або іншій моделі готовий сценарний матеріал.

Системний промпт для генерації (SYSTEM_PROMPT_GENERATION) створювався для сценарію, де модель має діяти як незалежний креативний автор, отримуючи на вхід такий самий неструктурований контекст, як і експерименті аналізу. Його мета – максимально активувати творчі й стилістичні здібності моделі.

У промпті є такі вимоги:

– використовувати «художній, образний стиль, а не технічний опис», що є критичною для перетворення сухих фактів структурованого контексту на живу прозу;

– вимога щодо «живих, правдоподібних діалогів», що розкривають характери, змушувала модель інтегрувати в мову і дії персонажів їхні риси, визначені в контексті;

– вимога завжди завершувати сцену «логічною крапкою», що бореться з поширеною проблемою обірваних генерацій, змушуючи модель створювати відчуття цілісності та виконаної мікрозадачі.

3.3.2 Сценарії

Сценарії формують набір вхідних даних для тестування великих мовних моделей. Кожен сценарій був структурований у форматі контексту, запиту користувача і еталонного виводу.

Сценарії охопили різні жанри (технофентезі, військова стратегія, нео-нуар) та пропонували моделям складні завдання, що вимагають не лише прямого використання фактів, але й інтеграції психологічних рис персонажів у діалоги та внутрішні монологи.

Короткі описи кожного розробленого сценарію (повні версії сценаріїв наведено в додатку Б):

– сценарій 1: «Ефірний Контрабандист» (технофентезі).

Контекст: Світ Арія та плавуче місто Зоряний Порт, де магія (Ефір) призводить до небезпечної «кришталізації» людей. Головні персонажі: Ліам Скайвокер (цинічний Ефірний контрабандист), Ейла (холодна, логічна архітекторка Ефірних схем), яка прагне помститися системі Паладіїв Світла. Конфлікт: Вони володіють рідкісним Кришталем Стабілізації, який Ліам хоче продати (гроші), а Ейла – знищити (безпека та помста за сестру). Паладії переслідують Ліама.

Запит: сцена зустрічі Ліама та Ейли у схованці, де Ліам, щойно втік від патруля, напружено обговорює з логічною Ейлою долю Кришталю. Акцент на контрасті характерів та описі Ефірних схем.

Мета тестування: перевірка здатності моделей інтегрувати складні правила магічного світу, відобразити психологічний контраст і напруженість у діалозі (цинізм проти логіки) і підтримувати вказаний користувачем стиль і жанр;

– сценарій 2: «Полководець і Магія Крові» (військове фентезі).

Контекст: військовий табір у гірському проході Тінь Дракона. Головний персонаж: Лорд Каель – молодий полководець, який ненавидить магію, але керується стратегією «повільного оточення». Має шрам і носить Амулет Корвуса. Конфлікт: Каель готується до облоги фортеці «Мертва Скеля», яку утримує Лорд-Чаклун Зейн.

Запит: сцена, де Каель отримує звістку про те, що Зейн укріпив фортецю магією крові. Каель має відреагувати цинічно на згадку про магію, проявити свою швидкість рішень, але згадати свою фірмову стратегію.

Мета тестування: оцінка здатності моделей відобразити внутрішній конфлікт (ненависть до магії проти необхідності перемоги), використати символічні деталі (шрам, амулет) і точно відтворити стратегічні військові терміни, зберігаючи при цьому цинічний тон;

– сценарій 3: «Невловимий і Годинник Фараона» (жанр нео-нуар).

Контекст: Місто 1978 року, таверна «Тихий Притулок». Головні персонажі: Віктор «Невловимий» Коваль – ветеран-інженер, контрабандист із власним «Кодексом» (без насильства/наркотиків) і глибоким песимізмом. Має молодшу сестру Ліду. Загроза: комісар Іван Драганов, котрий прагне знищити репутацію Віктора. Конфлікт: Віктор має терміново вивезти викрадену національну реліквію «Годинник Фараона», але отримує звістку про аварію Ліди.

Запит: сцена в підсобці, де Віктор розбирає радіоприймач і отримує погані новини. Акцент на цинізмі Віктора, його механічній винахідливості й гострому внутрішньому конфлікті між бізнесом (Годинник) і сімейним обов’язком (Ліда).

Мета тестування: перевірка здатності моделей точно інтегрувати механічні деталі (опис радіоприймача), відобразити внутрішній монолог та створити сцену високої напруги, де сімейний борг переважає над професійним ризиком.

3.3.3 Отримані результати і оцінка аналізу і генерації

Після того, як відповіді з аналізом по сценаріях від моделей отримано, проводиться оцінка. Самі результати аналізу (тексти) наведені в додатку В. Автоматичні метрики підраховуються застосунком, а оцінку по метриці Actions Logic надають експерти, що ознайомилися зі сценаріями, вихідними текстами моделей, зокрема, з частиною про можливі дії персонажів в сцені.

На рисунку 3.2 зображено скріншот роботи застосунку щодо підрахунку оцінок щодо результатів аналізу.

Аналогічно проводиться експеримент із генерацією: моделі видають художні тексти, наведені в додатку В. Із ними ознайомлюються експерти і дають свої оцінки.

```
🔹 Обробка аналізу ID=1 | Сценарій=1, Модель=GPT-4o
Введіть оцінки логічності дій (через кому, 1–10): 9
Середня оцінка логічності дій: 0.9
 - Збережено: coverage=0.886, halluc=0.326, F1=0.818, actions=0.900, Q=0.824

🔹 Обробка аналізу ID=2 | Сценарій=1, Модель=Claude 4 Opus
Введіть оцінки логічності дій (через кому, 1–10): 10
Середня оцінка логічності дій: 1.0
 - Збережено: coverage=0.909, halluc=0.192, F1=0.800, actions=1.000, Q=0.889

🔹 Обробка аналізу ID=3 | Сценарій=1, Модель=Gemini 2.5 Pro
Введіть оцінки логічності дій (через кому, 1–10): 9.5
Середня оцінка логічності дій: 0.95
 - Збережено: coverage=0.932, halluc=0.283, F1=0.621, actions=0.950, Q=0.821

🔹 Обробка аналізу ID=4 | Сценарій=2, Модель=GPT-4o
Введіть оцінки логічності дій (через кому, 1–10): 9
Середня оцінка логічності дій: 0.9
 - Збережено: coverage=1.000, halluc=0.065, F1=0.941, actions=0.900, Q=0.942

🔹 Обробка аналізу ID=5 | Сценарій=2, Модель=Claude 4 Opus
Введіть оцінки логічності дій (через кому, 1–10): 9
Середня оцінка логічності дій: 0.9
 - Збережено: coverage=1.000, halluc=0.037, F1=0.947, actions=0.900, Q=0.950

🔹 Обробка аналізу ID=6 | Сценарій=2, Модель=Gemini 2.5 Pro
Введіть оцінки логічності дій (через кому, 1–10): 9
Середня оцінка логічності дій: 0.9
 - Збережено: coverage=1.000, halluc=0.126, F1=0.833, actions=0.900, Q=0.905

🔹 Обробка аналізу ID=7 | Сценарій=3, Модель=GPT-4o
Введіть оцінки логічності дій (через кому, 1–10): 8.5
Середня оцінка логічності дій: 0.85
 - Збережено: coverage=0.967, halluc=0.405, F1=0.571, actions=0.850, Q=0.760

🔹 Обробка аналізу ID=8 | Сценарій=3, Модель=Claude 4 Opus
Введіть оцінки логічності дій (через кому, 1–10): 9
Середня оцінка логічності дій: 0.9
 - Збережено: coverage=0.967, halluc=0.085, F1=0.667, actions=0.900, Q=0.874

🔹 Обробка аналізу ID=9 | Сценарій=3, Модель=Gemini 2.5 Pro
Введіть оцінки логічності дій (через кому, 1–10): 9
Середня оцінка логічності дій: 0.9
 - Збережено: coverage=0.967, halluc=0.250, F1=0.500, actions=0.900, Q=0.799
```

Рисунок 3.2 – Робота застосунку щодо підрахунку оцінок аналізу

На рисунках 3.3–3.5 зображено скріншот роботи застосунку щодо збору оцінок експертів і підрахунку фінальних оцінок моделей по всім сценаріям.

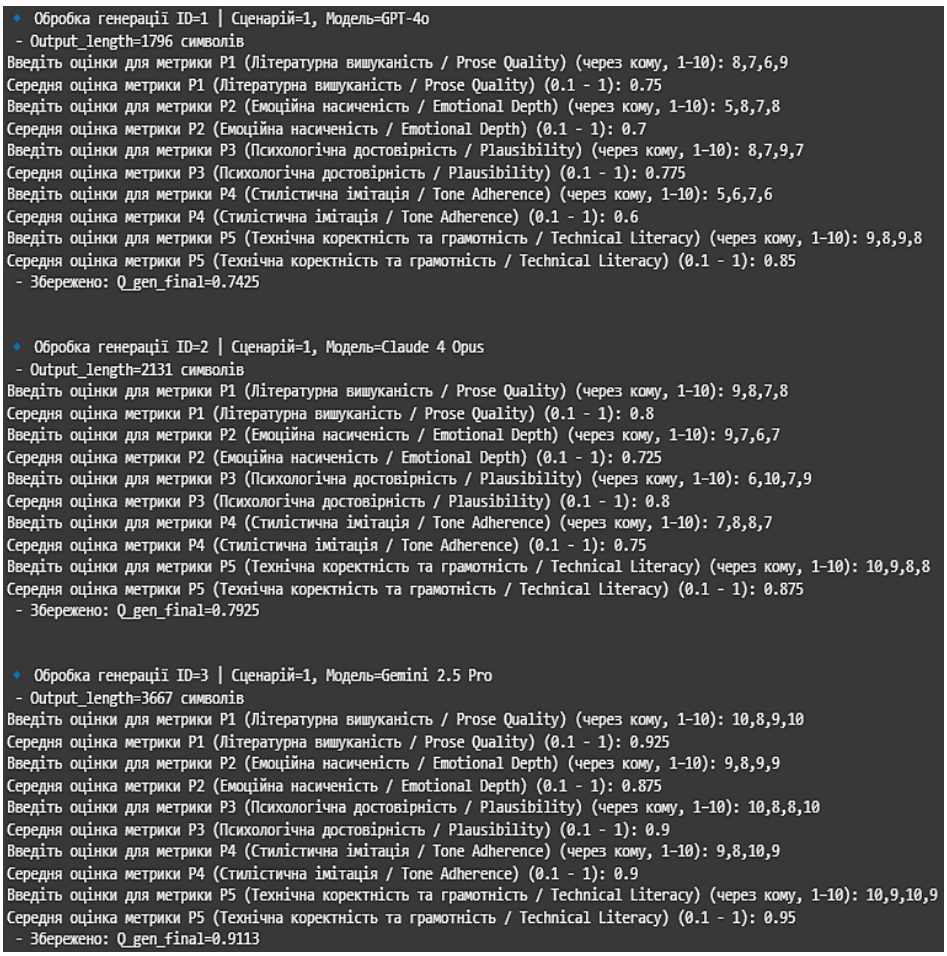

Рисунок 3.3 – Робота застосунку щодо підрахунку оцінок генерації (сценарій 1)

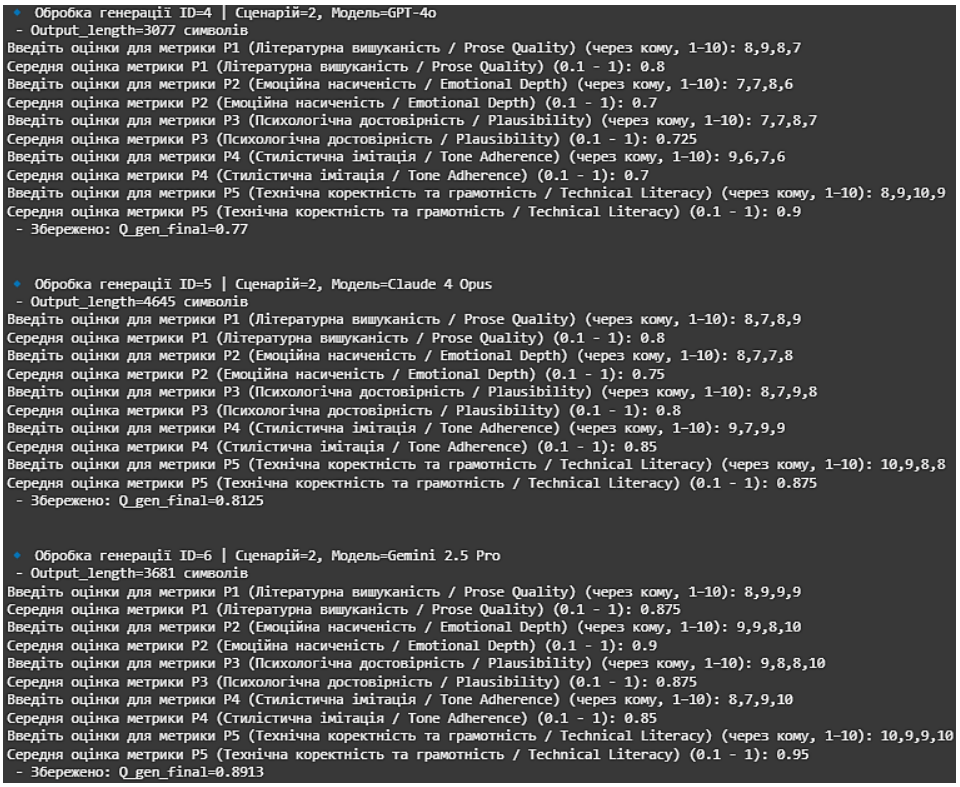

Рисунок 3.4 – Робота застосунку щодо підрахунку оцінок генерації (сценарій 2)

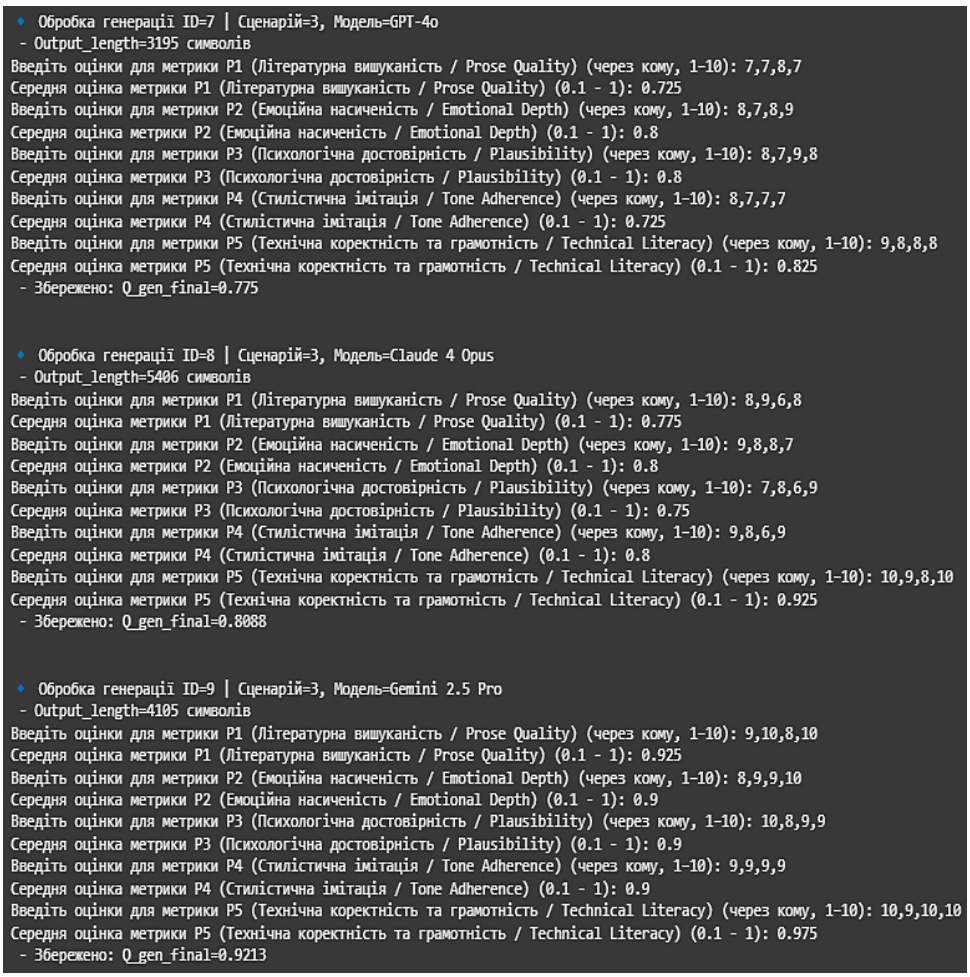

Рисунок 3.5 – Робота застосунку щодо підрахунку оцінок генерації (сценарій 3)

3.4 Порівняння досліджених великих мовних моделей для аналізу та генерації художніх текстів

3.4.1 Порівняння моделей в завданні аналізу

Результати тестування трьох моделей у завданні аналізу на трьох різних за жанром та складністю сценаріях представлені у таблиці 3.1.

Отримані результати демонструють, що всі три системи мають високу якість відтворення змісту: середній показник Coverage перевищує 0,95, що свідчить про повне чи майже повне охоплення ключових аспектів вихідного тексту.

Проте характер помилок і рівень стабільності значно різняться між моделями.

Таблиця 3.1 – Результати тестування моделей у задачі аналізу на всіх сценаріях

Модель	**Сценарій**	**Coverage**	**Hallucination Rate**	$F1$	**Actions Logic**	Q_A
GPT-4o	1	0,89	0,33	0,82	0,90	0,82
Claude 4 Opus	1	0,91	0,19	0,80	1,00	0,89
Gemini 2.5 Pro	1	0,93	0,28	0,62	0,95	0,82
GPT-4o	2	1,00	0,07	0,94	0,90	0,94
Claude 4 Opus	2	1,00	0,04	0,95	0,90	0,95
Gemini 2.5 Pro	2	1,00	0,17	0,83	0,90	0,91
GPT-4o	3	0,97	0,41	0,57	0,85	0,76
Claude 4 Opus	3	0,97	0,09	0,67	0,90	0,87
Gemini 2.5 Pro	3	0,97	0,25	0,50	0,90	0,80

Модель Claude 4 Opus продемонструвала найвищу загальну ефективність, отримавши середній інтегральний показник $Q_A \approx 0{,}91$, що вище за інші системи. Її сильними сторонами стали винятково низький рівень галюцинацій (0,1) та висока точність у визначенні іменованих сутностей ($F1$ = 0,81). Вона зберігає логічну цілісність навіть у складних сценаріях, формулює висновки чітко й стримано, не схильна до вигадування додаткових деталей. Таким чином, Claude можна охарактеризувати як модель із «критичним мисленням».

На рисунку 3.6 наведено порівняння метрик для Claude 4 Opus.

GPT-4o показала порівняно стабільні результати, утримуючи баланс між логікою та повнотою, проте мала вищий рівень галюцинацій (0,27). Це може свідчити про її тенденцію до більш творчої інтерпретації тексту. Незважаючи на це, середній рівень точності й узгодженості залишився високим ($Q_A \approx 0{,}84$), а здатність зберігати смислову структуру – практично бездоганною.

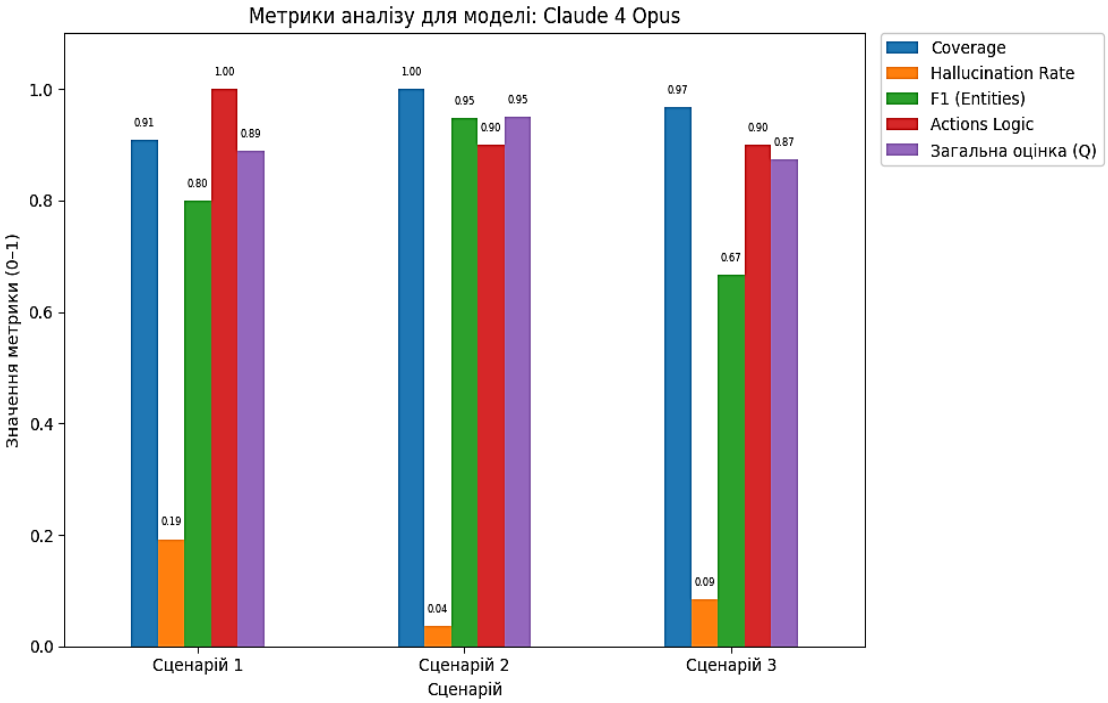

Рисунок 3.6 – Графік порівняння метрик аналізу для Claude 4 Opus

Модель видається орієнтованою на глибоке розуміння контексту, але з меншою фокусованістю на суворій фактологічній точності.

На рисунку 3.7 наведено порівняння метрик для GPT-4o.

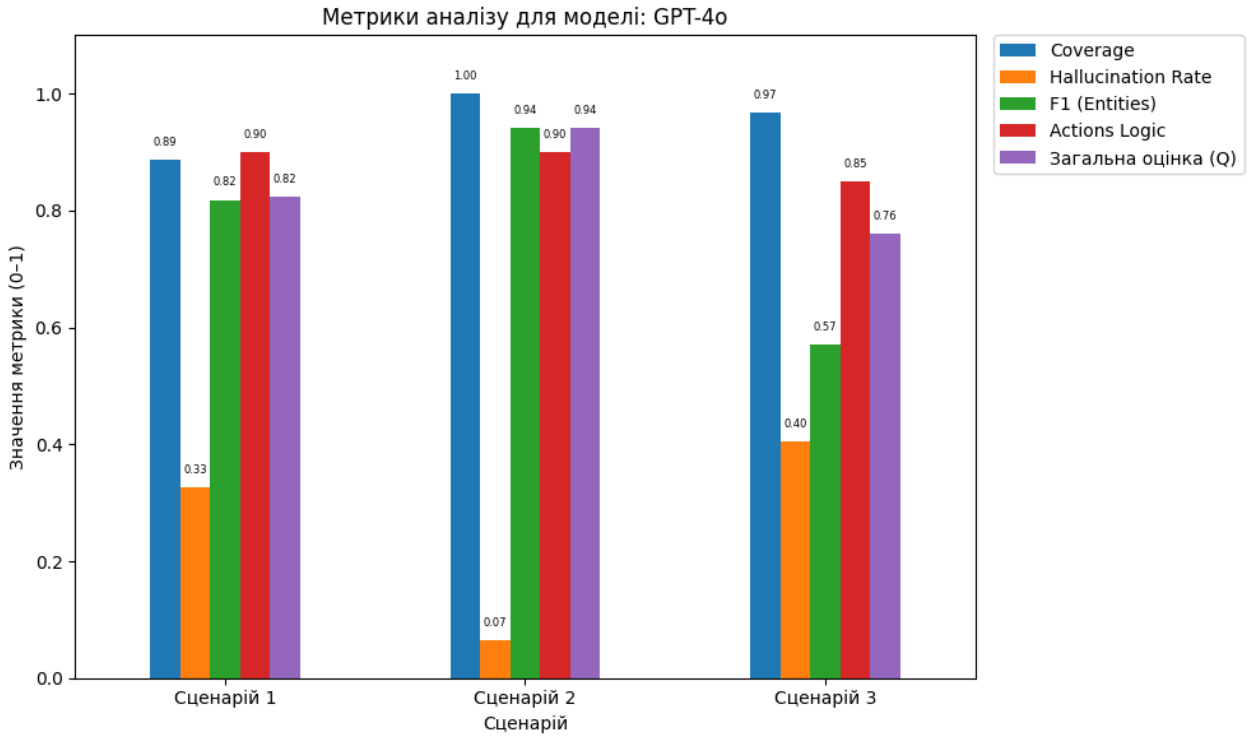

Рисунок 3.7 – Графік порівняння метрик аналізу для GPT-4o

Gemini 2.5 Pro продемонструвала найвище охоплення змісту (Coverage ≈ 0,97), проте мала нижчий показник точності сутностей ($F1 \approx 0{,}65$) і дещо вищу схильність до генерації неточних тверджень. Попри це, загальна якість залишилася на рівні інших моделей ($Q_A \approx 0{,}84$). Gemini добре справляється з узагальненням та логічними висновками, але поступається конкурентам у глибині розбору деталей і точності міжтекстових зв'язків.

На рисунку 3.8 наведено порівняння метрик для Gemini 2.5 Pro.

Всі моделі продемонстрували високий рівень когнітивної узгодженості у виконанні аналітичного завдання. Водночас Claude 4 Opus виявилася найнадійнішою й найпослідовнішою системою в аспектах логічної побудови суджень і мінімізації фактологічних спотворень.

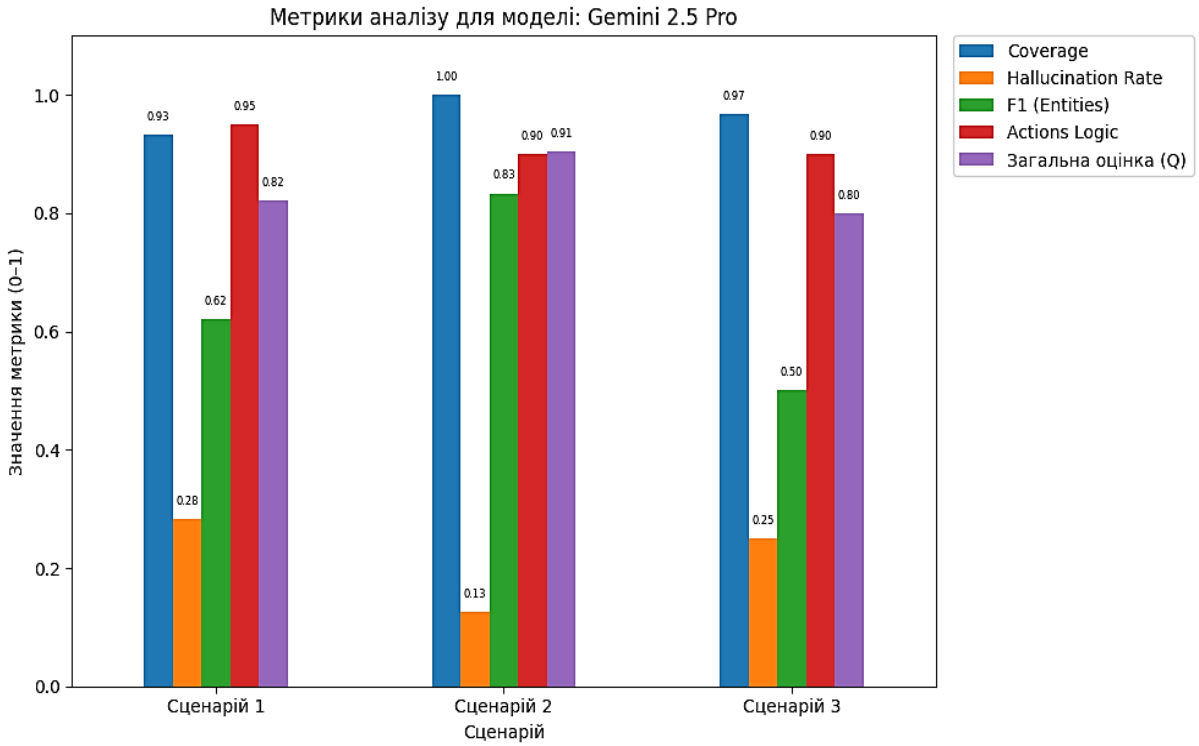

Рисунок 3.8 – Графік порівняння метрик аналізу для Gemini 2.5 Pro

GPT-4o вирізняється більш гнучким і варіативним мисленням, що робить її цінною для інтерпретаційних і творчих типів аналізу, тоді як Gemini 2.5 Pro демонструє потенціал у системному охопленні контексту, але потребує вдосконалення в точності деталізації.

У завданні аналітичної обробки художніх текстів найвищу якість продемонструвала Claude 4 Opus, що дозволяє розглядати її як оптимальну основу для подальшого комбінованого підходу до аналізу та генерації.

3.4.2 Порівняння моделей в завданні генерації

Результати тестування трьох моделей у завданні генерації на трьох різних за жанром і складністю сценаріях представлені у таблиці 3.2.

Таблиця 3.2 – Результати тестування моделей у задачі генерації на всіх сценаріях

Модель	**Сценарій**	$\boldsymbol{P_1}$	$\boldsymbol{P_2}$	$\boldsymbol{P_3}$	$\boldsymbol{P_4}$	$\boldsymbol{P_5}$	$\boldsymbol{Q_G}$
GPT-4o	1	0,75	0,70	0,78	0,60	0,80	0,74
Claude 4 Opus	1	0,80	0,73	0,80	0,75	0,88	0,79
Gemini 2.5 Pro	1	0,93	0,88	0,90	0,90	0,95	0,91
GPT-4o	2	0,80	0,70	0,73	0,70	0,90	0,77
Claude 4 Opus	2	0,80	0,75	0,80	0,85	0,88	0,81
Gemini 2.5 Pro	2	0,88	0,90	0,88	0,85	0,95	0,89
GPT-4o	3	0,73	0,80	0,80	0,73	0,83	0,78
Claude 4 Opus	3	0,78	0,80	0,75	0,80	0,93	0,81
Gemini 2.5 Pro	3	0,93	0,90	0,90	0,90	0,98	0,92

Результати тестування в таблиці 3.2 дозволяють розглянути якість генерації крізь призму п’яти аспектів: літературної вишуканості (P_1), емоційної насиченості (P_2), психологічної достовірності (P_3), стилістичної імітації (P_4) та технічної грамотності (P_5).

Підсумковий індикатор якості генерації Q_G узагальнює ці виміри і дає змогу порівняти моделі між собою у кожному конкретному сценарії.

При першому погляді помітна перевага Gemini 2.5 Pro у всіх трьох сценаріях: її Q_G коливається в межах від 0,89 до 0,92, що вказує на те, що ця модель послідовно поєднує образну мову з граматичною точністю й логічною цілісністю сюжетів.

За показником літературній вишуканості Gemini показує вищі значення (0,88–0,93) порівняно з Claude (приблизно 0,78–0,8) і GPT-4o (приблизно 0,73–0,8). Це означає, що Gemini генерує багатші мовні конструкції, більшу різноманітність синтаксису та більш образні формулювання. Така перевага помітна в тих сценаріях, де від тексту вимагається художня виразність і робота зі стилістичними фігурами – відзначена висока P_1 Gemini прямо корелює з її високими Q_G-показниками.

У показнику емоційної насиченості Gemini також лідирує, демонструючи значення 0,88–0,9. Це показує, що модель краще передає емоційні відтінки сцени і викликає більш виразний емоційний відгук у рецензентів. Claude демонструє середні результати з $P_3 \approx$ 0,73–0,8 – модель вміє створювати емоційні моменти, але робить це менш послідовно, а GPT-4o в цілому має нижчі значення P_2 (приблизно 0,7–0,8), що вказує на іноді більш «суху», менш емоційно насичену манеру викладу.

Показник психологічної достовірності, здатності вчинків і діалогів бути логічно обґрунтованими, також корелює з інтегральною якістю: Gemini має P_3 на рівні 0,88–0,9, Claude – близько 0,75–0,8, GPT-4o — в діапазоні 0,73–0,8. Високі значення P_3 у Gemini свідчать про те, що її тексти рідше містять нелогічні мотиви або невмотивовані вчинки персонажів; у Claude і GPT-4o іноді реєструються дрібні розбіжності у мотиваціях, що знижує загальну правдоподібність сцен. З практичної точки зору це означає: якщо важлива внутрішня узгодженість поведінки героїв та суворе дотримання встановлених правил світу, Gemini у дослідах поводить себе найкраще.

Стосовно стилістичної імітації, тобто наскільки текст відповідає заданому жанру і тональним вимогам, спостерігається явна слабкість GPT-4o у першому сценарії (P_4 = 0,6), тоді як Claude і Gemini дають значно вищі результати (Claude: 0,75–0,85, Gemini: 0,85–0,9).

Це вказує, що GPT-4o менш надійно слідує тональним вказівкам запиту, іноді «відходячи» від жанрових або тональних обмежень. Навпаки, Gemini проявляє стабільність у відтворенні жанру і завдання, а Claude інколи ближче до Gemini, інколи – дещо поступається.

Технічна коректність P_5 найчастіше виявляється менш проблемною для всіх трьох моделей, але і тут Gemini показує найвищі значення (0,95–0,98), Claude – високі, але трохи нижчі (0,875–0,93), а GPT-4o – прийнятні (0,8–0,9). Це означає, що помилки орфографії, пунктуації та несумісності фактичних деталей найрідше зустрічаються в текстах Gemini. Високе значення P_5 у Gemini підсилює загальне враження якісної генерації, оскільки технічна грамотність безпосередньо впливає на сприйняття художньої глибини й достовірності. Її можна розглядати в якості моделі-генератора в методі комбінування моделей.

Порівняння метрик генерації для Gemini 2.5 Pro подано на рисунку 3.9, для Claude 4 Opus – на рисунку 3.10, для GPT-4o – на рисунку 3.11.

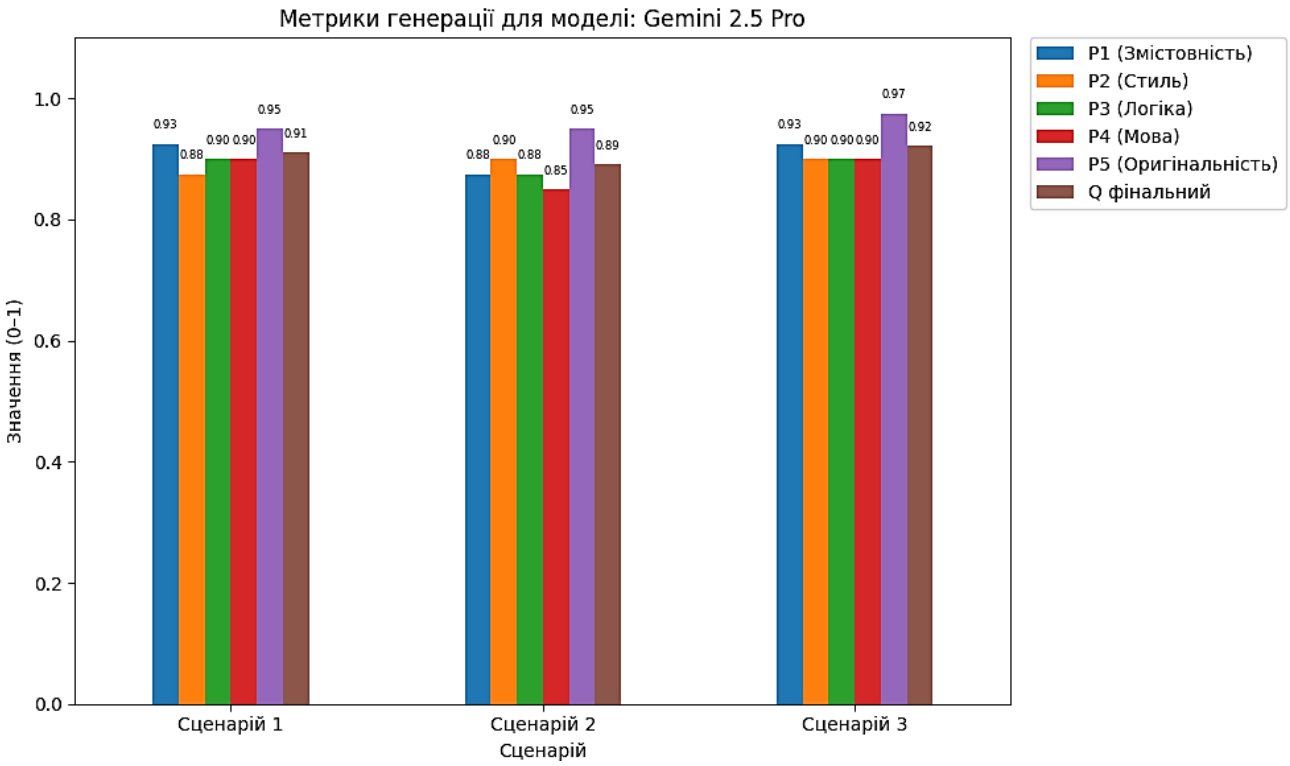

Рисунок 3.9 – Графік порівняння метрик генерації для Gemini 2.5 Pro

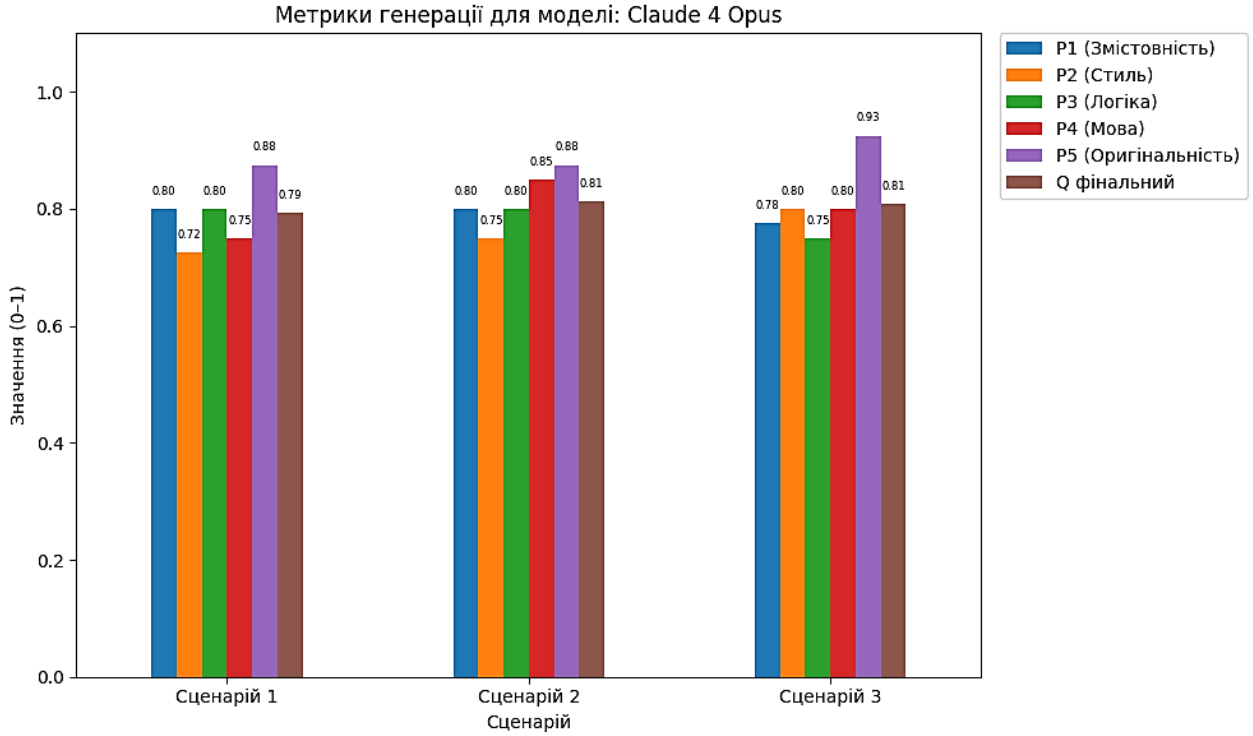

Рисунок 3.10 – Графік порівняння метрик генерації для Claude 4 Opus

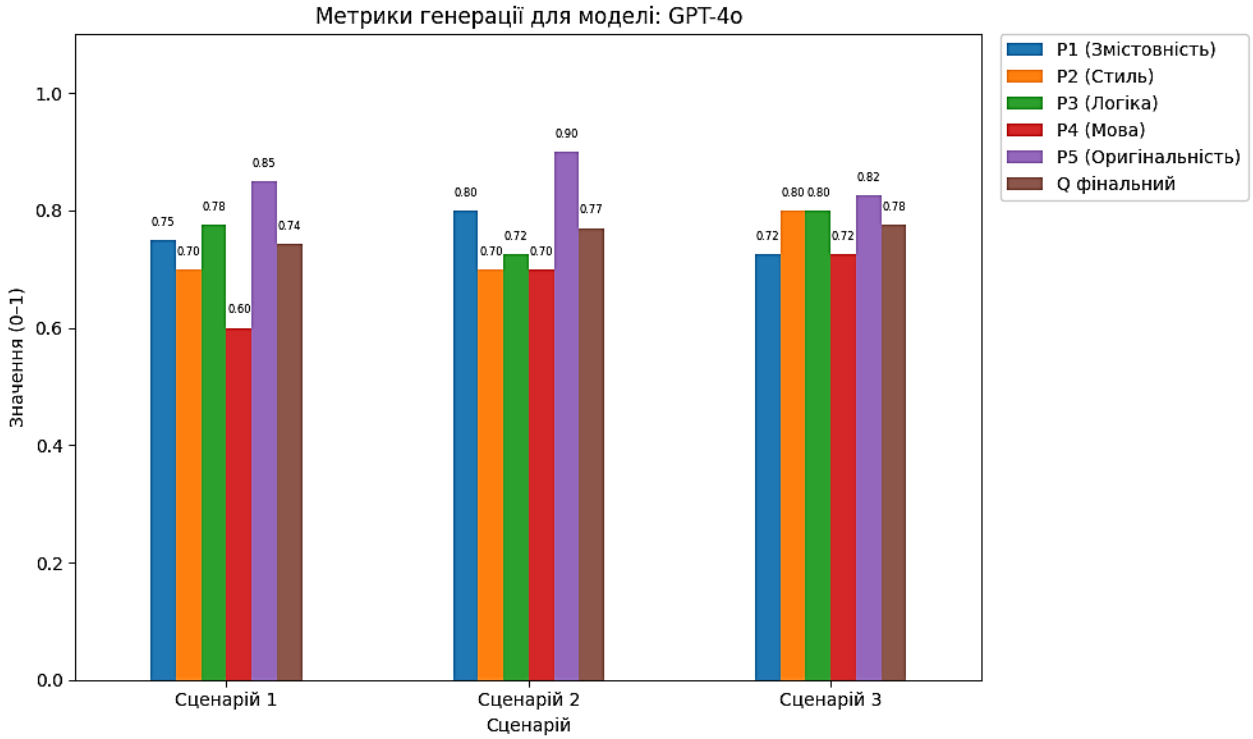

Рисунок 3.11 – Графік порівняння метрик генерації для GPT-4o

3.4.3 Оцінка роботи комбінованого методу та порівняння із результатами моделей окремо

У процесі оцінки продуктивності моделей на завданнях аналізу та генерації виявлено, що жодна з розглянутих мовних моделей не демонструє стабільного лідерства одночасно в обох типах завдань.

Claude 4 Opus продемонструвала найкращі результати в аналітичних задачах, зокрема у структуризації сюжетних елементів, виділенні тематичних акцентів і послідовності логічних ланцюгів. Gemini 2.5 Pro виявилася ефективнішою у генеративних сценаріях, особливо в аспектах стилістичної цілісності, природності мовлення та художньої виразності.

Виходячи з цього, тестується комбінований метод, що інтегрує сильні сторони обох моделей. У межах методу модель-аналітик (Claude 4 Opus) спочатку виконує детальний структурний аналіз вхідного контексту, формуючи когерентний опис ключових фактів, персонажів і сюжетних вузлів.

Далі цей аналіз подається як систематизований промпт моделі-генератору (Gemini 2.5 Pro), що дозволяє їй спиратися не на неструктурований текст користувача, а на логічно впорядковану базу знань про сцену чи фрагмент твору. Такий підхід дає змогу оцінити, наскільки попередня аналітична обробка покращує якість художньої генерації. Результати генерації методом комбінування моделей наведено у додатку В.

На рисунку 3.12 зображено скріншот роботи застосунку збору оцінок експертів і підрахунку фінальних оцінок щодо роботи методу.

Оцінки генерації методом комбінування моделей подані у таблиці 3.3.

Таблиця 3.3 – Оцінки генерації методом комбінування моделей

Сценарій	$\boldsymbol{P_1}$	$\boldsymbol{P_2}$	$\boldsymbol{P_3}$	$\boldsymbol{P_4}$	$\boldsymbol{P_5}$	$\boldsymbol{Q_G}$
1	0,95	0,93	0,95	0,93	0,95	0,94
2	0,98	0,95	0,95	0,98	0,95	0,96
3	0,98	0,93	0,98	0,98	0,98	0,97

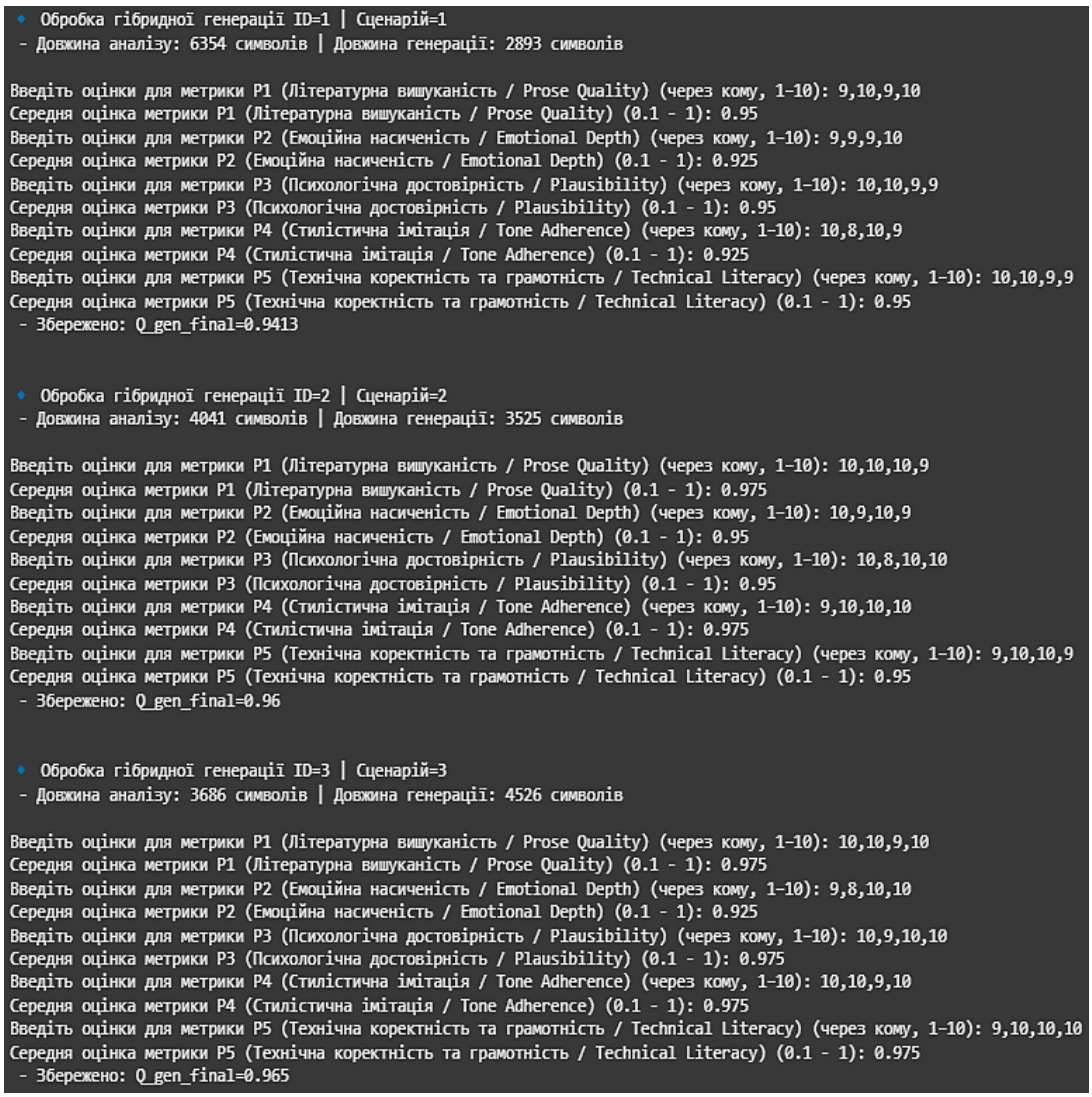

Рисунок 3.12 – Робота застосунку щодо підрахунку оцінок генерації методом комбінування моделей

Як видно із таблиці 3.3, у всіх трьох сценаріях комбінований метод продемонстрував середній інтегральний показник Q_H вище 0,94, що перевищує результати окремих моделей у тих самих умовах. Найвищий показник 0,97 отримано у третьому сценарії, де контекст мав складну багаторівневу структуру та вимагав збереження емоційної цілісності сцени. Це свідчить про здатність підходу комбінування оптимізувати аналітичну основу і покращувати творчий аспект тексту, утворюючи баланс між когнітивною точністю та художньою глибиною. Комбінований підхід забезпечив кращу узгодженість і логічну зв’язність створених текстів, значне зменшення кількості смислових розривів, повторів і стилістичних невідповідностей.

Графік оцінок генерації комбінованого методу подано на рисунку 3.13.

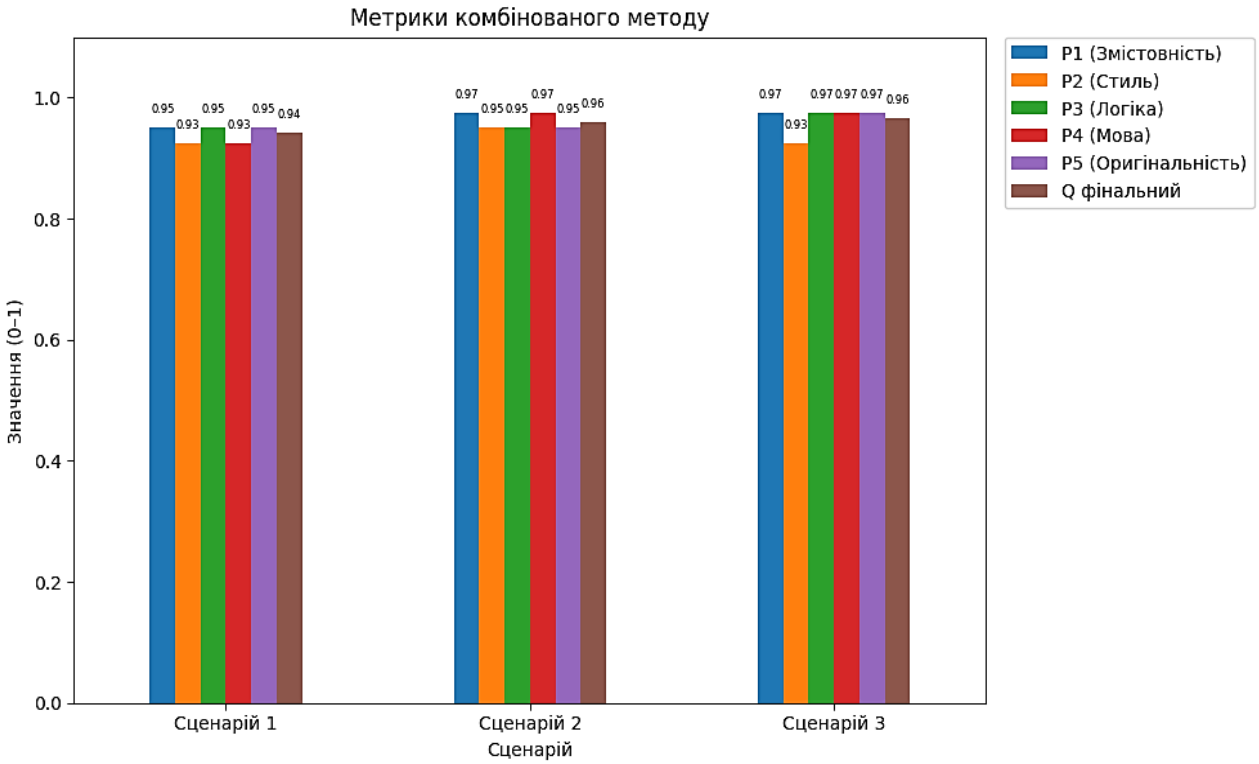

Рисунок 3.13 – Графік оцінок генерації комбінованого методу

3.4.4 Порівняння результатів комбінованого методу із результатами моделей

У таблиці 3.4 наведено порівняння результатів генерації моделей GPT-4o, Claude 4 Opus, Gemini 2.5 Pro і комбінованого методу.

Як видно із таблиці 3.4, фінальна оцінка Q_G результатів комбінованого методу виявилася кращою за оцінки моделей у всіх сценаріях. Також комбінований метод обійшов моделі по всіх критеріях, отримавши однакову оцінку лише у критерії технічної грамотності (P_5) із моделлю Gemini 2.5 Pro.

Такий результат пояснюється тим, що моделлю-генератором в методі власне і була ця модель – вона проявила свої «знання» орфографії і пунктуації у методі (на що модель-аналітик і її наданий аналіз сценарію не впливають).

Таблиця 3.4 – Порівняння результатів генерації моделей GPT-4o, Claude 4 Opus, Gemini 2.5 Pro і комбінованого методу

Сценарій	Метрика	GPT-4o	Claude 4 Opus	Gemini 2.5 Pro	Комбіно-ваний метод
1	P_1	0,75	0,8	0,93	0,95
	P_2	0,70	0,73	0,83	0,93
	P_3	0,78	0,80	0,90	0,95
	P_4	0,60	0,75	0,90	0,93
	P_5	0,80	0,88	0,95	0,95
	Q_G	0,74	0,79	0,91	0,94
2	P_1	0,80	0,80	0,88	0,98
	P_2	0,70	0,75	0,90	0,95
	P_3	0,73	0,80	0,88	0,95
	P_4	0,70	0,85	0,85	0,98
	P_5	0,90	0,88	0,95	0,95
	Q_G	0,77	0,81	0,89	0,96
3	P_1	0,73	0,78	0,93	0,98
	P_2	0,80	0,80	0,90	0,93
	P_3	0,80	0,75	0,90	0,98
	P_4	0,73	0,80	0,90	0,98
	P_5	0,83	0,93	0,98	0,98
	Q_G	0,78	0,81	0,92	0,97

3.5 Перспективи подальшої роботи

Результати, отримані у рамках порівняльного аналізу великих мовних моделей та тестування комбінованого методу, демонструють значний потенціал інтеграції аналітичної точності та художньої генерації.

Водночас проведене дослідження висвітлило низку аспектів, які потребують подальшого вивчення та вдосконалення.

Інтеграція моделей-аналітиків та генераторів може бути розширена шляхом багаторівневої композиції, де кілька моделей-аналітиків взаємодіють між собою, створюючи багатошарову, структуровану репрезентацію тексту. Такий підхід дозволить глибше оцінювати когнітивні та емоційні аспекти сцен, зменшуючи ризик логічних невідповідностей і підвищуючи достовірність художніх описів. Також варто звернути увагу на адаптивну генерацію на основі жанрових та стилістичних параметрів.

У майбутніх дослідженнях можна розробити механізми автоматичного налаштування моделей під конкретні жанрові рамки чи художні вимоги, що дозволить моделювати тон, емоційний відтінок та мовні особливості тексту більш точно. Перспективним є поєднання комбінованого методу з інтерактивними підходами, де користувач або експерт може коригувати проміжні аналітичні представлення, впливаючи на кінцеву генерацію тексту. Це не лише підвищить точність і узгодженість контенту, а й відкриє нові можливості для дослідження гібридних систем у творчих процесах.

Варто дослідити масштабування підхід на більш широкий спектр сценаріїв і мов, включаючи багатомовні тексти, складні міжособистісні взаємодії й динамічні сюжетні лінії. Розширення бази сценаріїв дозволить оцінювати гнучкість моделей та їх здатність зберігати якість у різноманітних контекстах. Майбутні роботи можуть включати інтеграцію з інструментами автоматичної оцінки якості тексту, такими як семантична узгодженість, крос-сценарний аналіз персонажів і емоційна когерентність, що дозволить не лише кількісно оцінювати результати, а й формувати більш повну картину потенціалу великих мовних моделей у літературній творчості.

ВИСНОВКИ

Таким чином, у кваліфікаційній роботі досліджено великі мовні моделі для аналізу та генерації художніх текстів.

Вирішено такі завдання:

– проведено огляд сучасних великих мовних моделей і методів оцінки їхньої ефективності у завданнях аналізу та генерації текстів, що дало можливість виявити сильні й слабкі сторони існуючих моделей;

– створено промпти і тестові сценарії різного жанру і рівня детальності для всебічного дослідження моделей, що дозволило оцінити їхню продуктивність у різноманітних контекстах;

– проведено експериментальне тестування трьох обраних моделей на завданнях аналізу, що дало змогу визначити модель-лідерку за показниками охоплення, точності виявлення сутностей, логіки дій та комбінованої оцінки;

– проведено тестування моделей у завданні генерації художніх текстів, оцінюючи їх за критеріями емоційної насиченості, психологічної достовірності, стилістичної імітації й технічної грамотності, що дозволило визначити оптимальну модель для генерації;

– розроблено та протестовано метод комбінування моделей, де результати моделі-аналітика використовувалися як структурована база для моделі-генератора, що продемонструвало підвищення якості створених текстів порівняно з окремими моделями;

– проведено порівняльний аналіз результатів окремих моделей і гібридного методу, що підтвердило потенціал розробленого підходу для подальшого використання в завданнях творчої генерації тексту.

У рамках роботи проведено дослідження моделей GPT-4o, Claude 4 Opus і Gemini 2.5 Pro. Створено програмний застосунок для тестування ефективності цих моделей та методу комбінування моделей на практичних сценаріях, в якому на основі вхідних контекстів і запитів від моделей отримується структурований

аналіз тексту і генерація художніх сцен, а також проводиться автоматична й експертна оцінка якості результатів. Результати дослідження представлені у вигляді таблиць, графіків і детального опису показників.

Наукова новизна роботи полягає у запропонованому комбінованому методі, у якому функції аналізу та генерації розділяються між двома моделями, що найкраще проявили себе в одній з двох задач. Такий метод враховує сильні сторони окремих моделей та забезпечує підвищення якості кінцевого результату.

Результати роботи апробовано у вигляді статті у науковому журналі «International Journal of Academic and Applied Research» [45] і 2 тез доповідей під час IV Міжнародної науково-практичної конференції «Technologies, theories and developments: modern scientific teaching» [46] і II Міжнародної науково-практичної студентської конференції «ІТ-простір сьогодення: тенденції, інновації та перспективи розвитку» [47].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Franceschelli, G., & Musolesi, M. (2024). On the creativity of large language models. AI & SOCIETY, 1-11.

2. Bommasani, R. (2021). On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258.

3. Achiam, Josh, et al. (2023) "Gpt-4 technical report." arXiv preprint arXiv:2303.08774.

4. Team, G., Georgiev, P., Lei, V. I., Burnell, R., Bai, L., Gulati, A., ... & Batsaikhan, B. O. (2024). Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. arXiv preprint arXiv:2403.05530.

5. Valmeekam, K., Marquez, M., Sreedharan, S., & Kambhampati, S. (2023). On the planning abilities of large language models-a critical investigation. Advances in Neural Information Processing Systems, 36, 75993-76005.

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

7. Microsoft. Elevating experiences with AI, from productivity to personalization. URL: https://www.microsoft.com/en-us/dynamics-365/blog/business-leader/2024/08/29/elevating-experiences-with-ai-from-productivity-to-personalization/ (дата звернення 29.09.2025).

8. I saw the first major 'AI game' coming to PC, and it convinced me of its potential for storytelling. URL: https://www.pcgamer.com/hidden-door-ai-game-narrative-rpg/ (дата звернення 16.08.2025).

9. Gómez-Rodríguez, C., & Williams, P. (2023). A confederacy of models: A comprehensive evaluation of LLMs on creative writing. arXiv preprint arXiv:2310.08433.

10. Wan, Q., Hu, S., Zhang, Y., Wang, P., Wen, B., & Lu, Z. (2024). " It Felt Like Having a Second Mind": Investigating Human-AI Co-creativity in Prewriting with Large Language Models. Proceedings of the ACM on human-computer interaction, 8(CSCW1), 1-26.

11. Bellemare-Pepin, A., Lespinasse, F., Thölke, P., Harel, Y., Mathewson, K., Olson, J. A., ... & Jerbi, K. (2024). Divergent creativity in humans and large language models. arXiv.

12. Ismayilzada, M., Stevenson, C., & van der Plas, L. (2024). Evaluating creative short story generation in humans and large language models. arXiv preprint arXiv:2411.02316.

13. Shanahan, M., & Clarke, C. (2023). Evaluating large language model creativity from a literary perspective. arXiv preprint arXiv:2312.03746.

14. Marco, G., Rello, L., & Gonzalo, J. (2024). Small language models can outperform humans in short creative writing: A study comparing slms with humans and llms. arXiv preprint arXiv:2409.11547.

15. Subbiah, M., Zhang, S., Chilton, L. B., & McKeown, K. (2024). Reading subtext: Evaluating large language models on short story summarization with writers. Transactions of the Association for Computational Linguistics, 12, 1290-1310.

16. Zhao, Y., Zhang, R., Li, W., & Li, L. (2025). Assessing and understanding creativity in large language models. Machine Intelligence Research, 22(3), 417-436.

17. Li, R., Zhu, C., Xu, B., Wang, X., & Mao, Z. (2025). Automated Creativity Evaluation for Large Language Models: A Reference-Based Approach. arXiv preprint arXiv:2504.15784.

18. Rane, N., Choudhary, S., & Rane, J. (2024). Gemini versus ChatGPT: applications, performance, architecture, capabilities, and implementation. Journal of Applied Artificial Intelligence, 5(1), 69-93.

19. Akter, S. N., Yu, Z., Muhamed, A., Ou, T., Bäuerle, A., Cabrera, Á. A., ... & Neubig, G. (2023). An in-depth look at gemini's language abilities. arXiv preprint arXiv:2312.11444.

20. Tu, L., Meng, R., Joty, S., Zhou, Y., & Yavuz, S. (2024). Investigating Factuality in Long-Form Text Generation: The Roles of Self-Known and Self-Unknown. arXiv preprint arXiv:2411.15993.

21. Garrido-Merchán, E. C., Arroyo-Barrigüete, J. L., & Gozalo-Brizuela, R. (2023). Simulating HP Lovecraft horror literature with the ChatGPT large language model. arXiv preprint arXiv:2305.03429.

22. Творошенко, І.С. (2021). Технології прийняття рішень в інформаційних системах: навч. посібник. *Харків: ХНУРЕ*.

23. Tvoroshenko I., Gorokhovatskyi V., Kobylin O., and Tvoroshenko A. (2023) Application of deep learning methods for recognizing and classifying culinary dishes in images, *International Journal of Academic and Applied Research*, 7(9), pp. 57-70.

24. Кобилін, О.А., & Творошенко, І.С. (2021). Методи цифрової обробки зображень: навч. посібник. *Харків: ХНУРЕ*.

25. Gorokhovatskyi V., Chmutov Y., Tvoroshenko I., and Kobylin O. (2025) Reducing computational costs by compressing the structural description in image classification methods, *Advanced Information Systems*, vol. 9, no. 1, pp. 5-12.

26. Pomazan, V., Tvoroshenko, I., and Gorokhovatskyi, V. (2023). Development of an application for recognizing emotions using convolutional neural networks, *International Journal of Academic Information Systems Research*, 7(7), pp. 25-36.

27. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2024) Improving the effectiveness of image classification structural methods by compressing the description according to the information content criterion, *Computers, Materials & Continua*, vol. 80, no. 2, pp. 3085-3106.

28. Gorokhovatskyi V., Tvoroshenko I. (2023) Identification of visual objects by the search request. *Int. scientific symp. «Intelligent Solutions-S». Computational intelligence. Decision making theory: proceedings of the international symposium, September 28, 2023*, Kyiv-Uzhorod, Ukraine, 25-27.

29. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., and Hudáková M. (2025) Image description compression in classification structural methods, *IEEE Access*, vol. 13, pp. 43631-43641.

30. Yakovleva O., Matúšová S., Tvoroshenko I., and Isaiev Y. (2024) Visitor counting based on video stream analysis from surveillance cameras to solve various business problems, *Verejná správa a regionálny rozvoj ekonómia, manažment a marketing*, XX(1), pp. 67-87.

31. Gorokhovatskyi, V., Tvoroshenko, I., Yakovleva, O., Hudáková, M., & Gorokhovatskyi, O. (2024). Application a committee of Kohonen neural networks to training of image classifier based on description of descriptors set. *IEEE Access*, *12*, 73376-73385.

32. Гороховатський В.О., Творошенко І.С. (2022) Аналіз багатовимірних даних за описом у формі множини компонент: монографія. Харків: ХНУРЕ, 124 с.

33. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2022) Cluster representation of the structural description of images for effective classification, *Computers, Materials & Continua*, 73(3), pp. 6069-6084.

34. Gorokhovatskyi V., Tvoroshenko I., and Yakovleva O. (2024) Transforming image descriptions as a set of descriptors to construct classification features, *Indonesian Journal of Electrical Engineering and Computer Science*, 33(1), pp. 113-125.

35. Daradkeh, Y. I., Gorokhovatskyi, V., Tvoroshenko, I., & Zeghid, M. (2022). Tools for fast metric data search in structural methods for image classification, *IEEE Access*, vol. 10, pp. 124738-124746.

36. Гороховатський В., Передрій О., Творошенко І., Марков Т. (2023) Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень, *Сучасні інформаційні системи*, 7(1), С. 5-13.

37. Gorokhovatskyi, V., & Tvoroshenko, I. (2024). An effective method for transforming an image description into a compact vector for classification.

38. Gorokhovatskyi, V., Tvoroshenko, I., Kobylin, O., & Vlasenko, N. (2023). Search for visual objects by request in the form of a cluster representation for the structural image description, *Advances in Electrical and Electronic Engineering*, 21(1), pp. 19-27.

39. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., Gadetska S., and Al-Dhaifallah M. (2023) Statistical data analysis models for determining the relevance of structural image descriptions, *IEEE Access*, vol. 11, pp. 126938-126949.

40. Tvoroshenko I., Pomazan V., Gorokhovatskyi V., and Kobylin O. (2023) Application of video data classification models using convolutional neural networks, *International Journal of Academic and Applied Research*, 7(11), pp. 134-145.

41. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Al-Dhaifallah M. (2022) Classification of Images Based on a System of Hierarchical Features, *Computers, Materials & Continua*, 72(1), pp. 1785-1797.

42. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Handwritten character recognition models based on convolutional neural networks, *International Journal of Academic Engineering Research*, 7(9), pp. 64-72.

43. Гороховатський В.О., Творошенко І.С., Чмутов Ю.В. (2022) Застосування систем ортогональних функцій для формування простору ознак у методах класифікації зображень, *Сучасні інформаційні системи*, 6(3), С. 5-12.

44. Bohdan N., Tvoroshenko I., Gorokhovatskyi V., and Kobylin O. (2025) Development of a hybrid method to enhance context memory for a chatbot application based on large language models, *International Journal of Academic Information Systems Research*, 9(10), pp. 7-18.

45. Suprun A., Tvoroshenko I., Gorokhovatskyi V., and Yakovleva O. (2025) Development and research of a method for the combined use of large language models for text generation, *International Journal of Academic and Applied Research*, 9(10), pp. 249-263.

46. Suprun A. (2025) Architectural features of modern large language models, *Abstracts of IV International scientific and practical conference «Technologies, theories and developments: modern scientific teaching», (September 23 – 26, 2025). Valencia, Spain,* pp. 23-26.

47. Suprun A. (2025) A combined method for leveraging large language models in the analysis and generation of creative texts, *II International Scientific and Practical Student Conference «IT Space Today: Trends, Innovations, and Development Prospects», (October 15, 2025),* pp. 306-309.