МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління

КРУШЕЛЬНИЦЬКИЙ Святослав Мар'янович

Інтелектуальна інформаційна система семантичного пошуку з використанням технології Retrieval-Augmented Generation (RAG) / Intelligent Information System for Semantic Search Using Retrieval-Augmented Generation (RAG)

Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Штучний інтелект

Кваліфікаційна робота

Виконав студент групи КНШІ-41
С. М. Крушельницький

Науковий керівник:
к.т.н., доцент, В. С. Коваль

Кваліфікаційну роботу допущено
до захисту
«___»_____ 2026 р.

Завідувач кафедри
_____ Н.В. Дзюбановська

Тернопіль – 2026

Факультет комп'ютерних інформаційних технологій
Кафедра інформаційно-обчислювальних систем і управління
Ступінь вищої освіти «бакалавр»
спеціальність 122 – Комп'ютерні науки
освітньо-професійна програма – Штучний інтелект

«ЗАТВЕРДЖУЮ»
В.о. завідувача кафедри ІОСУ
_____ Н.В. Дзюбановська
«__» _____ 2026 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
КРУШЕЛЬНИЦЬКОМУ Святослав Мар'яновичу
(прізвище, ім'я, по батькові)

1. Тема роботи: Інтелектуальна інформаційна система семантичного пошуку з використанням технології: Retrieval-Augmented Generation (RAG) / Intelligent Information System for Semantic Search Using Retrieval-Augmented Generation (RAG). керівник роботи В. С. Коваль к.т.н., доцент, доц. кафедри ІОСУ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом університету від 01грудня 2025 року № 894.
2. Строк подання кваліфікаційної роботи 25 травня 2026 р.
3. Вихідні дані до роботи завдання на кваліфікаційну роботу студента, наукові статті, технічна література.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
   - аналіз стану ринку корпоративних LLM у секторах BFSI та юридичного консалтингу, природа галюцинацій, обґрунтування RAG; аналіз еволюції від Naive RAG до Contextual Retrieval та Agentic RAG, проблематика «семантичної амнезії» та анафоричних зв'язків;
   - математичне забезпечення: механізм самоуваги, косинусна подібність, модель RAG-генерації, алгоритм HNSW, метрики RAGAS (context recall, context precision, faithfulness, answer correctness);
   - проєктування та реалізація Advanced RAG: синтетичні префікси моделлю gpt-5-nano, алгоритм «ковзного вікна» (k=3), стек, конвеєр індексації;
   - підготовка FinanceBench, порівняльний експеримент Naive RAG vs Advanced RAG за метриками RAGAS, верифікація гіпотези про перевагу контекстного збагачення над розширенням контекстного вікна LLM.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
   - схеми алгоритмів.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 01 грудня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Огляд літературних джерел відповідно до предметної області та складання плану роботи	до 01.01.2026 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.02.2026 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 15.03.2026 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05.2026 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2026 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів	до 25.05.2026 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту	до 30.05.2026 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 10.06.2026 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії	до 10.06.2026 р.	

Студент _____ Крушельницький С.М.
(підпис) (прізвище та ініціали)

Керівник кваліфікаційної роботи _____ Коваль В.С.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи: 75 с., 12 рис., 7 табл., 3 додатки, 37 джерел.

Об'єктом дослідження є процеси інтелектуального семантичного пошуку та вилучення даних зі складних корпоративних документів за допомогою великих мовних моделей.

Метою роботи є розроблення вдосконаленої інтелектуальної інформаційної системи семантичного пошуку (RAG-системи) на основі методу контекстного пошуку для мінімізації фактологічних «галюцинацій» та збереження семантичної цілісності даних.

Методи дослідження: технологія RAG, метод контекстного збагачення фрагментів (Contextual Retrieval) із використанням мовної моделі gpt-5-nano для синтезу описів, алгоритм «ковзного вікна» для збереження логічних зв'язків.

Результатом роботи є вдосконалена архітектура та програмний модуль RAG-системи, що усуває проблему семантичної фрагментації даних. Завдяки алгоритму ковзного вікна утримується зв'язок між суміжними фрагментами документа. Експериментальне тестування на наборі даних FinanceBench із використанням метрик RAGAS зафіксувало зростання повноти вилучення контексту на 35,46% і точності на 37,99%. Розроблений підхід гарантує повну трасованість кожної згенерованої відповіді до першоджерела при низьких обчислювальних витратах.

Розроблена система може бути використана для автоматизації процесів фінансового аудиту, юридичного консалтингу та інтеграції в промислові інформаційні екосистеми.

Ключові слова: ШТУЧНИЙ ІНТЕЛЕКТ, ВЕЛИКІ МОВНІ МОДЕЛІ, RAG, ПОШУКОВА ГЕНЕРАЦІЯ, КОНТЕКСТНИЙ ПОШУК, CONTEXTUAL RETRIEVAL, ОБРОБКА ПРИРОДНОЇ МОВИ, NLP, ФІНАНСОВИЙ АНАЛІЗ.

ANNOTATION

The bachelor’s thesis report: 75 pages, 12 figures, 7 tables, 3 appendices, 35 sources.

The object of the research is the processes of intelligent semantic search and data extraction from complex corporate documents using large language models.

The purpose of the work is to develop an advanced intelligent information system for semantic search (RAG system) based on the contextual retrieval method to minimize factual "hallucinations" and preserve the semantic integrity of data.

Research methods: RAG technology, Contextual Retrieval method using the gpt-5-nano language model for description synthesis, and a sliding window algorithm to preserve logical connections.

The result of the work is an advanced architecture and a software module of a RAG system that eliminates the problem of semantic data fragmentation. The sliding window algorithm maintains the connection between adjacent document fragments. Experimental testing on the FinanceBench dataset using RAGAS metrics confirmed an increase in context recall by 35.46% and context precision by 37.99%. The developed approach guarantees full traceability of each generated response to the original source at low computational costs.

The developed system can be used to automate financial auditing processes, legal consulting, and integration into industrial information ecosystems.

Keywords: ARTIFICIAL INTELLIGENCE, LARGE LANGUAGE MODELS, RAG, SEARCH-ASSISTED GENERATION, CONTEXTUAL SEARCH, CONTEXTUAL RETRIEVAL, NATURAL LANGUAGE PROCESSING, NLP, FINANCIAL ANALYSIS.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

LLM – Large Language Model

БД – база даних

RAG – Retrieval-Augmented Generation

КР – кваліфікаційна робота

NLP – Natural Language Processing

ШІ – штучний інтелект

MCP – Model Context Protocol

HNSW – Hierarchical Navigable Small World

TTFT – Time To First Token

TCO – Total Cost of Ownership

SS – Semantic Similarity

FS – Factual Similarity

TP – True Positives

FP – False Positives

FN – False Negatives

ВСТУП

Упродовж 2024 – 2026 років великі мовні моделі стали стандартним інструментом корпоративної роботи з неструктурованими даними. GPT-4, Claude, Gemini та їхні наступники синтезують і узагальнюють тексти на рівні, що часто не відрізнити від людського. Проблема в іншому: моделі впевнено генерують хибні факти – і ця їхня звичка особливо болюча там, де помилка коштує дорого. У медицині, фінансах, аудиті та юридичному консалтингу «галюцинація» – не технічний казус, а реальний ризик [3, 7, 22].

Параметрична пам'ять LLM фіксується в момент завершення навчання й далі не оновлюється [2]. Модель не бачить внутрішньої документації підприємства, тому генерує правдоподібні, але фактологічно хибні відповіді. RAG-архітектура знімає це обмеження: перед генерацією модель звертається до зовнішньої векторної бази й вилучає звідти актуальні факти [5]. Однак, базові RAG-системи (Naive RAG) у фінансовому аудиті зіткнулися з тим, що фрагментація тексту руйнує його семантичну цілісність [14, 15].

На початку 2026 року дослідницький фокус змістився на те, як саме інформація подається RAG-системі. Основний напрям – Advanced RAG [3, 11]: контекстне збагачення (Contextual Retrieval), гібридні архітектури, агентні системи [8, 16]. Моделі з надвеликим контекстним вікном не витіснили RAG, бо на базах знань обсягом у терабайти він обходиться дешевше й працює швидше – повні документи не потрібно завантажувати в контекст. Такий підхід відповідає «Концепції розвитку штучного інтелекту в Україні», яка ставить у пріоритет вдосконалення публічного управління та економічної безпеки [19].

Актуальність роботи пов'язана з проблемою «семантичної амнезії»: механічне розбиття звітів на сегменти розриває зв'язки між сутностями та числовими показниками [15, 17]. Контекстне збагачення на етапі індексації це компенсує – кожен фрагмент отримує програмний опис, і анафоричні посилання перестають бути проблемою. Gpt-5-nano при цьому дає точність вилучення до 39% у фінансових задачах проти 28% у базових системах, а коштує мінімально [6, 13].

Мета роботи – побудувати вдосконалений алгоритм RAG-системи на основі методу контекстного пошуку (Contextual Retrieval) для зменшення ризику фактологічних галюцинацій і підвищення точності вилучення даних зі складних корпоративних документів [8].

Для досягнення поставленої мети визначено наступні завдання:

– дослідити, як семантична фрагментація впливає на деградацію контексту в базових RAG-системах;

– розробити архітектуру Advanced RAG із використанням gpt-5-nano для синтезу описів фрагментів тексту;

– впровадити алгоритм «ковзного вікна» для збереження динамічної пам'яті про попередні сегменти й оптимізації обчислювальних витрат;

– оцінити підхід на FinanceBench за метриками RAGAS [4, 6, 12, 13];

– порівняти результати з Naive RAG і сформулювати висновки щодо ефективності методу.

Об'єктом дослідження є процес семантичного пошуку інформації у неструктурованих корпоративних документах інтелектуальними інформаційними системами на основі архітектури RAG.

Предмет дослідження – методи контекстного збагачення фрагментів тексту ситуативним макро-контекстом та їх програмна інтеграція в RAG-системи для підвищення точності вилучення знань [8, 10].

У роботі застосовано такі методи: глибоке навчання для розпізнавання закономірностей у текстових даних; контекстний пошук для збагачення фрагментів ситуативним макро-контекстом; алгоритм «ковзного вікна» для усунення семантичної амнезії; гібридний пошук, що поєднує семантичні вектори з лексичним; оцінювання за метриками RAGAS (Context Recall, Context Precision, Faithfulness) [4, 12, 30]. Математичний опис генерації префіксів і векторизації збагачених фрагментів наведено у розділах 2 і 3.

Наукове значення роботи – вдосконалення архітектури RAG через контекстне збагачення, яке розв'язує проблему анафори й зберігає семантичну цілісність тексту при фрагментації. Програмне додавання синтезованого макро-

контексту до кожного інформаційного вузла підвищує точність його представлення у латентному просторі [20]. Практичний результат – програмний модуль, що вилучає знання в реальному часі зі складних фінансових і юридичних документів. Кожна генерація трасується до першоджерела; тестування зафіксувало зниження рівня фактологічних помилок. На промислових обсягах даних система працює без деградації швидкодії й залишається дешевою в експлуатації.

Основні результати дослідження висвітлено у науковій публікації: Коваль В. С., Крушельницький С. М. «Підхід до побудови RAG-систем на основі удосконаленого методу контекстного пошуку», опублікованій у науковому фаховому виданні категорії «Б» – електронному журналі «Наука і техніка сьогодні». У роботі представлено вдосконалену архітектуру Advanced RAG, обґрунтовано застосування моделі gpt-5-nano та алгоритму «ковзного вікна» для збереження семантичної цілісності даних. Експериментальна оцінка на наборі FinanceBench за метриками фреймворку RAGAS зафіксувала зростання повноти вилучення контексту на 35,46% та точності на 37,99%.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Опис предметної області

У 2024–2026 роках LLM перейшли від експериментальних розробок до промислового застосування. Моделі GPT-4, Claude 3, Gemini, а згодом спеціалізовані рішення рівня gpt-5-nano досягли якості, достатньої для впровадження у задачі обробки знань у корпоративному секторі [26].

Глобальний ринок корпоративних мовних моделей демонструє стрімке зростання, що підтверджується динамікою капіталізації галузі. У 2025 році ринок оцінювався в 4,84 – 6,85 млрд доларів США, а прогнози на 2026 рік вказують на збільшення обсягів до 5,91 – 8,19 млрд доларів, з потенціалом досягнення позначки у 48,25 – 58,32 млрд доларів до 2034 року [26]. Основні показники розвитку ринку та прогнози щодо впровадження LLM систематизовано в таблиці 1.1. Така траєкторія свідчить про те, що LLM перетворилися з інструментів підвищення продуктивності окремих працівників на фундаментальний елемент цифрової інфраструктури підприємств [1].

Таблиця 1.1 – Порівняння динаміки ринку [26]

Показник розвитку ринку корпоративних LLM	2024 (Факт)	2025 (Оцінка)	2026 (Прогноз)
Обсяг ринку (млрд дол. США)	4.50	6.50 - 6.85	5.91 - 8.19
Рівень впровадження в організаціях (%)	55%	72% - 78%	88%
Частка сектору BFSI (%)	~50%	55%	58%
Інвестиції в RAG-системи (млрд дол. США)	-	12.8 (за 2023-25)	-

Сектор банківських, фінансових послуг та страхування (BFSI), а також юридичний консалтинг стали лідерами в адаптації цих технологій. Це зумовлено високою питомою вагою неструктурованих даних у цих галузях – від багатосторінкових фінансових звітів (10-K, 10-Q) до складних нормативних актів

та контрактів [26]. Аналіз показує, що близько 67% організацій по всьому світу вже інтегрували генеративний ШІ у свої робочі процеси, причому 23% з них масштабують агентні системи, здатні до автономного прийняття рішень [26].

Проте, широка експлуатація LLM у корпоративному середовищі, де точність та дотримання нормативних вимог є безальтернативними умовами, стикається з фундаментальним технічним бар'єром – схильністю моделей до фактологічних «галюцинацій». Галюцинації у цьому контексті визначаються як генерація контенту, що є лінгвістично зв'язним та граматично правильним, але фактологічно помилковим або не підкріпленим зовнішніми доказами.

Статистичні дані за 2025 – 2026 роки розкривають масштаб цієї проблеми: середній рівень галюцинацій у сучасних LLM при виконанні загальних завдань коливається від 15% до 52% [21]. У вузькоспеціалізованих доменах ризики ще вищі. Наприклад, у юридичній практиці середній показник галюцинацій становить 18,7%, а при генерації посилань на судові справи він може сягати 94% [21]. У фінансовому секторі рівень помилок оцінюється в 13,8%, що є критичним рівнем для систем, відповідальних за оцінку ризиків або аудит [21].

Математична природа галюцинацій закорінена в архітектурі трансформерів. LLM за своєю суттю є статистичними двигунами прогнозування наступного токена, а не базами знань у класичному розумінні. Вони оптимізовані для максимізації ймовірності послідовності символів, що часто призводить до вибору «правдоподібного» варіанта замість «фактично точного». Дослідження MIT 2025 року підтвердило, що моделі стають на 34% впевненішими у своїх твердженнях (використовуючи слова «безумовно», «без сумніву»), коли вони саме галюцинують, що робить такі помилки надзвичайно складними для виявлення без автоматизованої перевірки [5].

Головною причиною фактологічної неспроможності LLM у корпоративному секторі є обмеженість їхньої параметричної пам'яті. Знання моделі обмежені даними, на яких вона була навчена (knowledge cutoff), і вона не має прямого доступу до динамічної внутрішньої документації підприємства, яка постійно оновлюється. Це створює ситуацію, коли модель, намагаючись бути «корисною»

та «допомагати» користувачеві, починає видавати вигадані відповіді – заповнювати прогалини в знаннях статистично ймовірною, але вигаданою інформацією.

Для вирішення цієї проблеми виникла концепція RAG. RAG-системи діють як міст між потужними когнітивними здібностями LLM та актуальними, перевіреними даними корпоративних баз знань. Технологія розділяє знання системи на дві частини:

– параметричні знання: внутрішні ваги нейронної мережі, що відповідають за розуміння мови та логічне виведення;

– непараметричні знання: зовнішні індекси, що містять факти, вилучені з PDF-файлів, баз даних або API в реальному часі.

Впровадження RAG дозволяє забезпечити фактологічне обґрунтування відповідей моделі (grounding): замість того, щоб покладатися на власну пам'ять, система спочатку шукає релевантну інформацію у векторизованому сховищі, а потім використовує її як контекст для генерації відповіді. Це теоретично гарантує стовідсоткову трасованість кожного твердження до оригінального джерела, що є критичною вимогою для фінансового аудиту та юридичного консалтингу.

На початку 2026 року індустрія перейшла від базових архітектур (Naive RAG) до систем нового покоління [10]. Дослідження на наборі даних FinanceBench – першому відкритому еталоні для фінансового аналізу – продемонстрували, що просте додавання RAG не гарантує успіху [13]. Встановлено, що без спеціалізованої підготовки контексту моделі досягають лише 19% точності у складних фінансових питаннях, тоді як при наявності ідеального контексту («оракул») точність зростає до 89.3% [27]. Це зумовлює розрив у точності, що становить 70 відсотків, який є наслідком деградації контексту під час фрагментації даних.

Процес розбиття документів на дрібні фрагменти (chunking) призводить до втрати семантичних зв'язків. Виникає феномен «семантичної амнезії», коли вилучений системою фрагмент містить важливі цифри (наприклад, чистий прибуток), але не містить інформації про те, якої компанії чи року ці цифри стосуються, оскільки ця назва залишилася в іншому фрагменті. Це породжує

проблему «невидимого займенника», що спотворює результати семантичного пошуку [8, 17].

Для подолання цих викликів у 2025 – 2026 роках ключовим напрямом став розвиток методів контекстного збагачення (Contextual Retrieval). Використання надлегких та дешевих моделей, таких як gpt-5-nano, дозволяє програмно додавати до кожного фрагмента тексту стислий «макро-контекст» всього документа [25]. Це забезпечує збереження глобальної логіки документа навіть у межах ізольованих сегментів, що критично важливо для точності представлення даних у латентному векторному просторі.

Таким чином, предметна область дослідження охоплює методи та алгоритми побудови вдосконалених RAG-систем, які інтегрують передові алгоритми контекстуалізації, семантичної векторизації та динамічного управління пам'яттю для забезпечення найвищого рівня фактологічної точності у корпоративних інтелектуальних системах.

1.2 Огляд існуючих методів побудови систем пошуку на основі RAG

Сучасний розвиток інтелектуальних систем характеризується переходом до архітектур, здатних оперувати корпоративними знаннями в режимі реального часу, що зумовлено обмеженнями параметричної пам'яті LLM [2]. Фундаментальні засади RAG було закладено групою дослідників під керівництвом П. Льюїса у 2020 році [10], відтоді технологія стала стандартом для забезпечення фактологічної достовірності у системах на основі LLM [3]. Завдяки цій технології обсяги інвестицій у ринок корпоративних LLM-систем демонструють стрімке зростання [1, 26].

Класична архітектура Naive RAG базується на послідовному лінійному процесі «Retrieve-then-Generate» [10], концептуальну блок-схему якого представлено на рисунку 1.1. Експлуатація базових RAG-систем у високоточних доменах виявила критичні вразливості, насамперед – схильність до фактологічних «галюцинацій» [22]. Статистичні дослідження підтверджують, що моделі часто генерують лінгвістично правильні, але фактологічно хибні відповіді через нестачу

якісного зовнішнього контексту [5, 21]. Існуючі методи пом'якшення галюцинацій вказують на необхідність структурної оптимізації етапів вилучення та обробки інформації [7].

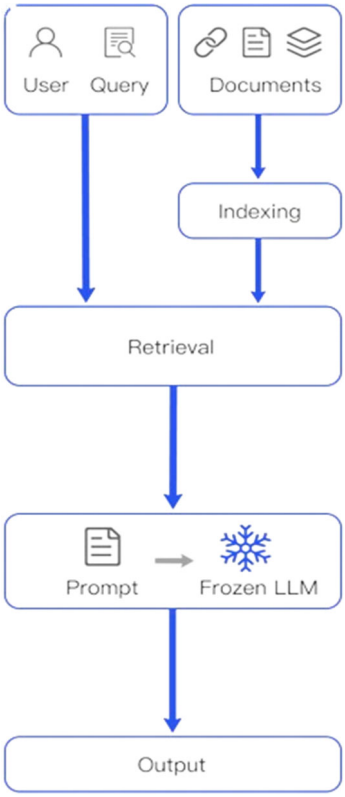

Рисунок 1.1 – Схема роботи Naive RAG [3]

Фундаментальним бар'єром для промислового застосування Naive RAG є деградація контексту під час підготовки даних. Ключовим етапом препроцесингу є chunking – розбиття тексту на сегменти фіксованого розміру [14]. Традиційні алгоритми (наприклад, RecursiveCharacterTextSplitter) здійснюють сегментацію без урахування логічних меж, що призводить до втрати контекстуальної цілісності [15]. Цей механічний підхід породжує феномен «семантичної амнезії» – втрату зв'язку між ізольованим фактом (наприклад, числовим показником) та його глобальними атрибутами (суб'єктом, часом, локацією) [17]. Для подолання цього обмеження застосовуються вдосконалені стратегії фрагментації тексту, що використовують рекурсивну семантику для збереження логічної структури документа [11, 14]. Якісна сегментація вимагає подальшого застосування

спеціалізованих моделей ембедингів речень для коректного перетворення тексту у векторний простір [20].

Архітектурним фундаментом таких моделей ембедингів є трансформер – нейронна мережа, що радикально змінила підходи до обробки природної мови завдяки механізму самоуваги (self-attention) [32]. На відміну від рекурентних нейронних мереж (RNN) та моделей довгої короткострокової пам'яті (LSTM), які опрацьовують дані послідовно, трансформери забезпечують паралельні обчислення, що надає їм масштабованість, необхідну для навчання на великих масивах корпоративних знань. Архітектуру трансформера представлено на рисунку 1.2.

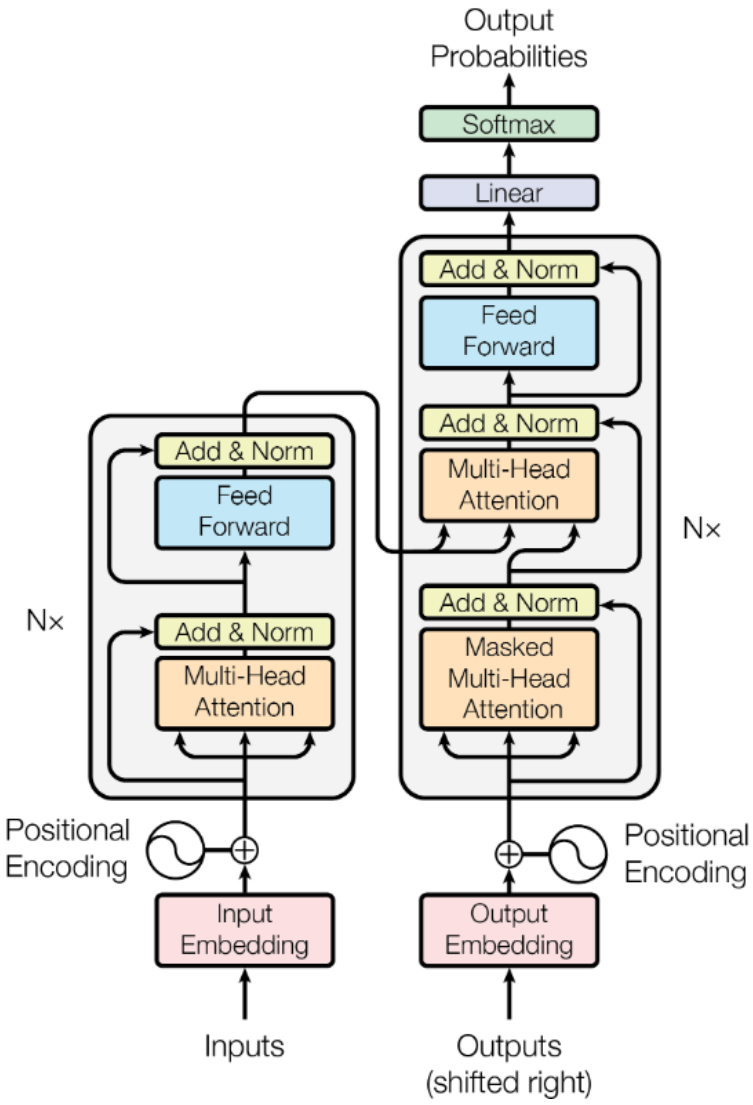

Рисунок 1.2 – Класична архітектура моделі Transformer [32]

Математичну основу self-attention становить взаємодія трьох матриць: запитів (Q), ключів (K) та значень (V). Для кожного вхідного токена розраховується оцінка релевантності відносно всіх інших токенів послідовності за допомогою скалярного добутку [32]:

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

де Q – матриця запитів, що представляє поточний контекст токенів;

K – матриця ключів, що відображає зв'язки між усіма токенами послідовності;

T – оператор транспонування матриці ключів;

d_k – розмірність векторів ключів, яка використовується як коефіцієнт масштабування для запобігання ефекту зникнення градієнтів;

V – матриця значень, яка містить вихідні векторні представлення токенів.

Функція softmax – процедура, що нормалізує отримані ваги, перетворюючи їх на розподіл ймовірності, де сума всіх значень дорівнює одиниці. Це дозволяє моделі динамічно визначати, на яких частинах вхідного контексту слід зосередитися для формування найбільш точного векторного представлення кожного слова.

Математична сутність семантичного пошуку в RAG-системах базується на обчисленні косинусної подібності між вектором запиту q та векторами документів d у латентному просторі [9]. Значення цієї метрики варіюється від −1 до 1, де 1 вказує на ідентичність напрямку векторів і інтерпретується як максимальна семантична схожість. У високоточних доменах, таких як фінанси, це дозволяє знаходити відповіді навіть тоді, коли термінологія запиту та документа не збігається буквально, але несе ідентичний смисл – наприклад, «чистий прибуток» та «net income». Водночас використання виключно косинусної подібності не вирішує проблему «невидимого займенника» (anaphora resolution) та феномен втрати релевантної інформації всередині контекстного вікна («Lost in the Middle»)

[11, 15, 28]. Технологічним стандартом подолання цих обмежень стало впровадження методу Contextual Retrieval (контекстного пошуку) [8, 16]. Метод передбачає генерацію ситуативного макро-префікса P_i для кожного фрагмента c_i перед векторизацією: $F_{final} = Embed(P_i \oplus c_i)$. Це дозволяє інтегрувати глобальний зміст документа в локальні контекстуальні ембединги, ефективно розв'язуючи анафори [18, 23].

Сучасна архітектура Advanced RAG обов'язково включає гібридний пошук (Hybrid Search), що об'єднує семантичний векторний пошук (Dense Retrieval) із точним лексичним пошуком алгоритмом BM25 (Lexical Retrieval) [16]. Структурну організацію та взаємодію модулів розширеного пошуку відображено на рисунку 1.3.

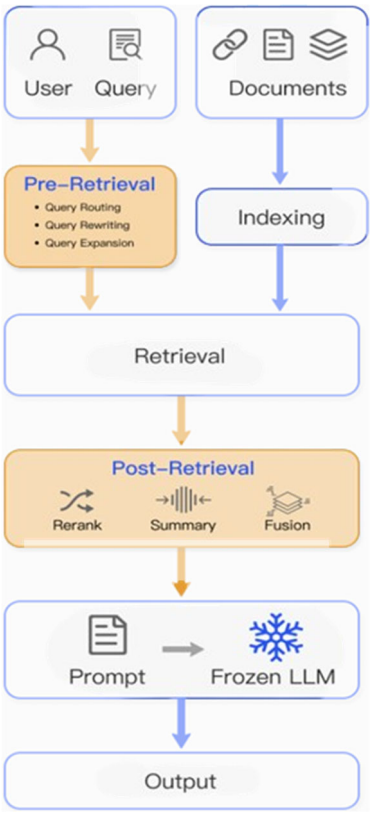

Рисунок 1.3 – Архітектура Advanced RAG із модулями гібридного пошуку та контекстуалізації [3]

Еволюційним розвитком технології став перехід від фіксованих лінійних конвеєрів до модульної архітектури (Modular RAG) [3]. На відміну від Advanced RAG, який лише оптимізує стандартні кроки пошуку та генерації, модульний підхід

забезпечує вищу гнучкість шляхом інтеграції нових, взаємозамінних структурних вузлів [3, 11]. Компоненти та можливі маршрути виконання у межах цієї парадигми систематизовано на рисунку 1.4.

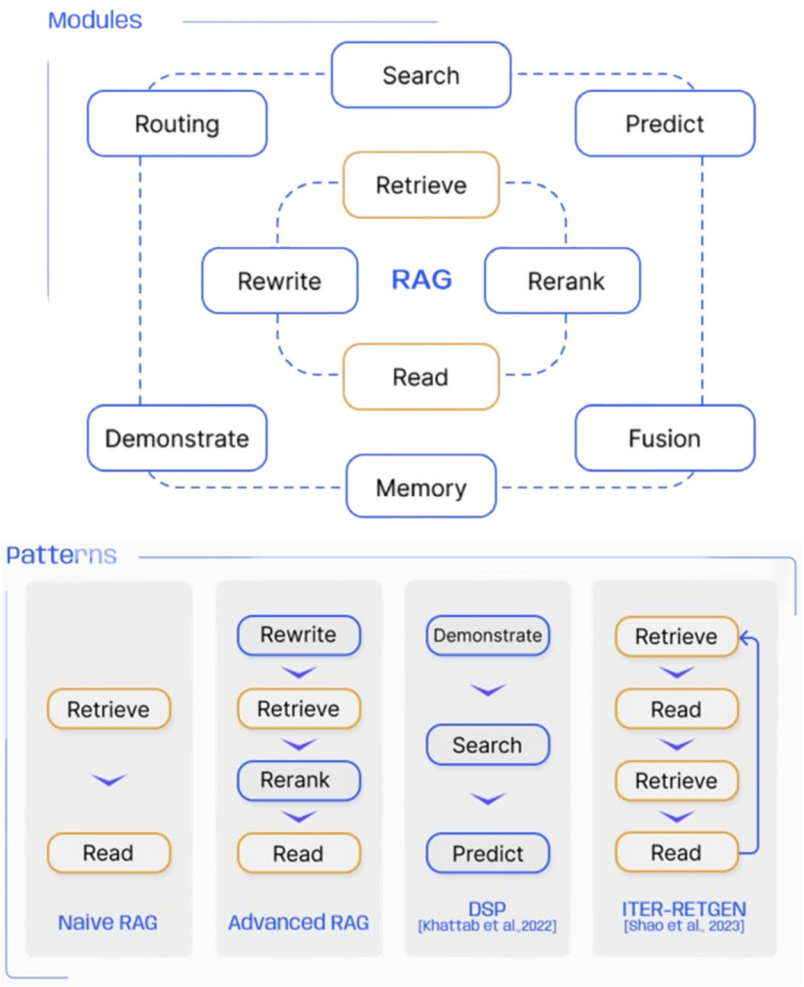

Рисунок 1.4 – Функціональна структура та інтеграція взаємозамінних вузлів у Modular RAG [3]

Основними компонентами цієї парадигми є:

– динамічна маршрутизація запитів: використання спеціалізованих модулів-роутерів (на базі легких класифікаторів або LLM) для аналізу наміру користувача та вибору оптимального сховища даних (векторного індексу, графа знань або реляційної БД) [3];

– трансформація та експансія запитів (Query Rewriting / Expansion): переформатування вхідного запиту для усунення лінгвістичної неоднозначності, що охоплює алгоритми генерації гіпотетичних відповідей (Hypothetical Document

Embeddings, HyDE) та декомпозицію складних багатокрокових запитів на підзапити [3, 11];

– модулі ітеративного та багатоетапного пошуку (Iterative / Multi-stage Retrieval): патерни, що реалізують послідовне уточнення контексту (наприклад, конвеєри Retrieve-Read-Retrieve), де результати попереднього кроку генерації стають основою для наступного пошукового запиту [3].

Наступним етапом еволюції є Agentic RAG (детальніше на рисунку 1.5), де агенти здатні до автономного ітеративного пошуку та самокорекції [23], комунікуючи з корпоративними базами даних через стандартизований інтерфейс MCP [31].

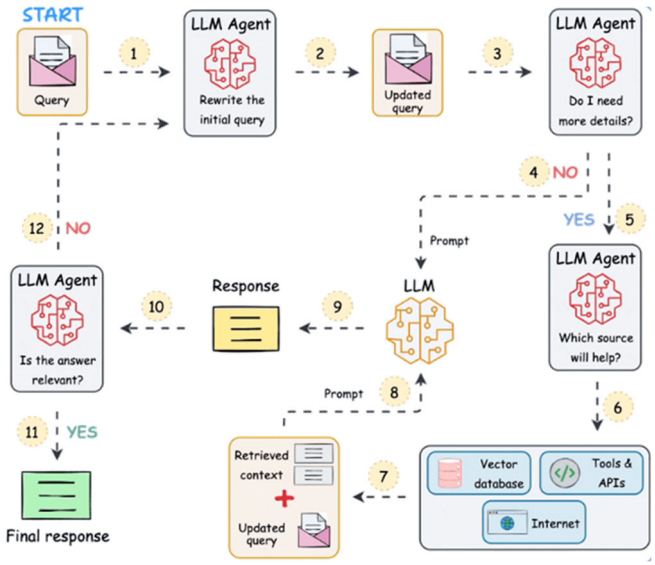

Рисунок 1.5 – Схема автономного конвеєра Agentic RAG із контурами ітеративного пошуку та самокорекції відповідей [24]

Альтернативним та доповнювальним підходом до адаптації моделей під специфіку конкретного домену є метод тонкого налаштування параметрів (Fine-tuning) [3, 7]. На відміну від RAG-архітектур, які динамічно використовують непараметричну зовнішню пам'ять, Fine-tuning модифікує безпосередньо внутрішні ваги нейронної мережі (параметричну пам'ять) через додаткове навчання на спеціалізованому корпусі текстів [2, 3]. Цей підхід забезпечує високу оптимізацію моделі під специфічні синтаксичні патерни, стилістику та вузькоспеціалізовану термінологію, проте має суттєві обмеження для систем

промислового пошуку: високу статичність знань після завершення навчання, значні обчислювальні витрати на регулярні цикли перенавчання та схильність до «непрозорих» галюцинацій (Black box) через неможливість прямого динамічного трасування джерел [3, 7, 22]. Детальне порівняння основних методів адаптації LLM до знань наведено у таблиці 1.2.

Таблиця 1.2 – Порівняння методів адаптації LLM до знань [11, 23, 25]

Критерій	Naive RAG	Advanced RAG	Modular RAG	Fine-tuning	Long Context (2M+)
Актуальність даних	Динамічна (Real-time)	Динамічна (Real-time)	Динамічна (Real-time)	Статична	Динамічна
Прозорість (Citations)	Часткова (ризик втрати логічних зв'язків)	100% Traceability	100% Traceability	Відсутня (Black box)	Присутня
Вартість запиту	Найнижча	Низька (Scale-ready)	Середня (через модульні виклики)	Висока	Дуже висока
Точність у вузьких доменах	Низька (через семантичну амнезію)	Найвища	Висока (ризик кумулят. помилки роутингу)	Середня	Середня (проблема Lost-in-middle)

Сукупність методів та алгоритмів, що утворюють математичний фундамент розглянутих архітектур RAG, ґрунтується на чотирьох взаємопов'язаних компонентах: контекстному кодуванні токенів через механізм self-attention трансформера, перетворенні тексту у векторні представлення (embeddings), оцінюванні семантичної близькості через косинусну подібність та ефективному індексуванні векторного простору алгоритмом HNSW. Систематизацію цих компонентів та їх ролі в архітектурі семантичного пошуку наведено в таблиці 1.3.

Таблиця 1.3 – Алгоритмічний базис систем семантичного пошуку

Компонент	Алгоритм	Роль у системі
Розуміння мови	Transformer Self-Attention	Обчислення контекстних ваг токенів
Векторизація	Dense Embeddings (d = 3072)	Перетворення тексту в латентний простір
Оцінка схожості	Cosine Similarity	Розрахунок семантичної відстані
Індексація вузлів	HNSW Graph	Ефективний пошук у великих масивах
Нормалізація	Softmax Layer	Формування ймовірнісних розподілів

Кожен із зазначених компонентів окремо є зрілою технологією, проте їх композиція в межах конвеєра Naive RAG не гарантує збереження семантичної цілісності документа на етапі індексації, що й зумовлює необхідність ширшого математичного аналізу деградації контексту, який проведено в наступному розділі.

Перехід від експериментальних систем до надійних корпоративних рішень [19] вимагає стандартизованої метрологічної бази. Для комплексного оцінювання ефективності архітектур пошукової генерації застосовується фреймворк RAGAS [4]. Цей інструментарій оцінює систему за метриками: Context Precision (точність ранжування), Context Recall (повнота вилучення фактів), Faithfulness (відсутність галюцинацій) та Answer Correctness (правильність синтезованої відповіді) [4, 12].

Для тестування розроблених систем у високоточних фінансових доменах використовується еталонний набір даних FinanceBench [6, 13, 27]. Оцінювання архітектур на базі таких стандартизованих наборів підтверджує, що впровадження контекстного збагачення (Advanced RAG) суттєво підвищує фактологічну точність порівняно з базовими методами [11].

Таким чином, аналіз існуючих підходів виявив три ключові обмеження базових RAG-архітектур: деградацію контексту через механічну фрагментацію документів, нездатність косинусної подібності розв'язувати анафоричні зв'язки та відсутність глобального ситуативного контексту в ізольованих векторних

представленнях. Зазначені недоліки підтверджують необхідність розробки вдосконаленої архітектури на основі методу Contextual Retrieval, яка усуває ефект «семантичної амнезії» шляхом програмного збагачення кожного фрагмента макро-контекстом документа на етапі індексації.

1.3 Постановка задачі дослідження

З аналізу предметної області випливає конкретна задача: контекст руйнується ще на етапі індексації – через механічне розбиття документів. Логіка проста: PDF або HTML нарізається на фрагменти фіксованого розміру без урахування змісту. Назва компанії залишається в одному фрагменті, чистий прибуток – в іншому. Вектор такого «осиротілого» числа в латентному просторі не несе в собі нічого, що дозволило б ретриверу його знайти за запитом користувача. Для вирішення поставленої задачі визначено наступні завдання дослідження:

– розробити метод послідовної контекстуалізації фрагментів тексту на основі концепції Contextual Retrieval [8] із використанням моделі gpt-5-nano для синтезу ситуативних макро-префіксів, що прив’язують кожен чанк до глобального змісту документа;

– реалізувати алгоритм «ковзного вікна» із буфером k=3 попередніх описів для збереження семантичної тяглості між суміжними сегментами документа та усунення анафоричних зв'язків;

– забезпечити структурний парсинг документів за допомогою технології Docling [29] зі збереженням ієрархічної структури таблиць і заголовків на етапі введення даних та обґрунтувати вибір компонентів векторного стеку;

– провести порівняльну експериментальну валідацію розробленої архітектури Advanced RAG щодо базової архітектури Naive RAG на наборі даних FinanceBench [6, 13] із використанням метрик фреймворку RAGAS [4] та верифікувати гіпотезу про перевагу контекстного збагачення над простим розширенням контекстного вікна мовної моделі.

Ключовою особливістю розробленого підходу є вдосконалення процесу підготовки даних для RAG-систем шляхом використання надлегкої моделі gpt-5-

nano як спеціалізованого контекстуалізатора. На відміну від існуючих підходів, що залучають повнорозмірні моделі для збагачення фрагментів [8, 16], запропоноване рішення дозволяє масштабувати технологію Contextual Retrieval на промислові обсяги даних без експоненційного зростання витрат. Важливим архітектурним доповненням системи також виступає алгоритм «ковзного вікна» із динамічним буфером описів, що забезпечує збереження семантичної зв'язності між суміжними сегментами.

Практичне значення роботи визначається створенням програмного модуля для вилучення знань у реальному часі зі складних фінансових і юридичних документів із повною трасованістю кожної згенерованої відповіді до першоджерела. Застосування моделі gpt-5-nano у поєднанні з механізмом Prompt Caching забезпечує економічну доступність розробленої системи для масштабування на бази знань корпоративного рівня.

Для практичної реалізації визначених завдань та розробки програмного модуля обрано технологічний стек, що відповідає вимогам продуктивності та масштабованості:

– модель контекстуалізації (gpt-5-nano) – обрана через унікальне поєднання когнітивних можливостей для синтезу описів та наднизької вартості, що робить технологію економічно життєздатною для терабайтних баз знань;

– модель ембедингів (text-embedding-3-large) – використовується для перетворення збагаченого тексту у вектори із підтримкою змінної розмірності, що дозволяє оптимізувати векторне сховище без суттєвої втрати точності;

– векторна база даних (Qdrant) – необхідна для зберігання та швидкого гібридного пошуку (векторний + BM25) за мільйонами збагачених векторів із підтримкою алгоритмів фільтрації у реальному часі;

– фреймворк розробки (LangChain) – забезпечує оркестрацію компонентів та управління ланцюжками викликів;

Обґрунтований вибір зазначених інструментів дозволяє реалізувати архітектуру, яка не просто шукає схожі слова, а розуміє контекст ситуації, що є критичним для точного аналізу фінансових звітів.

2 МЕТОДИ ТА АЛГОРИТМИ ПОБУДОВИ ВДОСКОНАЛЕНОГО RAG

2.1 Математичне забезпечення моделей та алгоритмів

Математичний апарат систем семантичного пошуку на основі RAG-архітектури охоплює чотири взаємопов'язані рівні: кодування токенів через механізм самоуваги трансформера, перетворення тексту у векторні представлення, оцінювання семантичної близькості та ефективна індексація векторного простору. Механізм самоуваги та функція косинусної подібності, що формують базис векторного представлення, детально розглянуті у підрозділі 1.2.

Класичний алгоритм Naive RAG реалізує двофазний конвеєр: попередня обробка документів на етапі індексації та пошук із генерацією під час виконання запиту. На фазі індексації документ розбивається алгоритмом RecursiveCharacterTextSplitter на фрагменти фіксованого розміру – типово 512 токенів – без урахування логічних меж тексту: розділів, абзаців чи речень. Кожен отриманий фрагмент незалежно векторизується моделлю ембедингів і зберігається у векторній базі даних. На фазі пошуку запит користувача перетворюється на вектор тим самим способом і порівнюється з усіма збереженими векторами за косинусною подібністю. Топ-k найближчих фрагментів передаються до LLM як контекст для генерації відповіді.

У Naive RAG є конкретне математичне обмеження: вектор будується виключно з тексту самого фрагмента. Назва компанії, звітний рік, валюта операції – якщо вони не потрапили в цей конкретний шматок, їх у векторі немає. Це породжує два ефекти, і обидва погані.

Перший – «семантична амнезія»: фрагмент може містити числовий показник (наприклад, «15.2 млрд дол.»), але не кодувати суб'єкт, до якого він відноситься, оскільки ця інформація залишилась в іншому фрагменті після механічного розбиття. Другий – «лексична колізія»: при обчисленні косинусної подібності фрагмент з аналогічними числовими показниками іншої компанії може отримати вищу схожість із запитом, ніж справді релевантний фрагмент. Обидва ефекти призводять до того, що навіть після успішного вилучення фрагмента генератор отримує неповний або хибний контекст і конфабулює відсутні атрибути.

Для подолання зазначених обмежень у вдосконаленій архітектурі Advanced RAG запроваджується проміжний етап між фрагментацією та векторизацією – контекстуальне збагачення. Замість векторизації ізольованого фрагмента застосовується операція конкатенації синтезованого ситуативного префікса P_i з оригінальним текстом:

$$F_{final}(i) = Embed(P_i \oplus c_i), \quad (2.1)$$

де P_i – ситуативний макро-префікс, синтезований моделлю gpt-5-nano для фрагмента c_i з урахуванням глобального контексту документа та буфера попередніх описів;

c_i – поточний текстовий фрагмент (чанк), для якого синтезується опис;

$\oplus$ – операція текстової конкатенації;

$F_{final}(i)$ – результуючий вектор збагаченого фрагмента у латентному просторі.

Порівняно з $e_i = Embed(c_i)$: формула (2.1) гарантує, що вектор $F_{final}(i)$ кодує не лише локальний зміст фрагмента, а й його глобальний ситуативний контекст – назву емітента, звітний рік, розділ документа. Це математично підвищує точність семантичного пошуку для релевантних фрагментів, усуваючи ефекти «семантичної амнезії» та «лексичної колізії». Детальний алгоритм синтезу префіксів P_i та механізм «ковзного вікна» для їх генерації описано у підрозділі 2.2.

Ключовим алгоритмом, що забезпечує швидкість роботи системи у промислових масштабах, є HNSW, реалізований у векторній базі даних Qdrant. HNSW будує багатошаровий граф, де нижній шар зберігає всі вузли-вектори з локальними зв'язками, а вищі шари містять лише підмножину вузлів із глобальними зв'язками для прискорення навігації. Пошук відбувається «згори вниз»: алгоритм жадібно переміщується до найближчих вузлів, поступово переходячи на нижчі, деталізовані шари. Математична складність пошуку становить $O(\log n)$, що дозволяє знаходити найрелевантніші фрагменти серед мільярдів записів за лічені мілісекунди. Якість апроксимації контролюється

параметрами m (кількість зв'язків на вузол) та ef (розмір динамічного списку кандидатів при побудові та пошуку в графі). Основні математичні моделі та алгоритмічні компоненти обох архітектур систематизовано в таблиці 1.3.

Для об'єктивного порівняння архітектур Naive RAG та Advanced RAG застосовується фреймворк RAGAS, який декомпозує якість системи на два незалежних виміри: точність ретривера та якість генератора. Фреймворк оцінює систему за чотирма метриками: Context Recall вимірює частку релевантних фактів, знайдених ретривером; Context Precision – якість впорядкування знайдених фрагментів; Faithfulness – ступінь відсутності галюцинацій у згенерованій відповіді; Answer Correctness – загальну відповідність відповіді еталону. Детальний математичний опис кожної метрики з відповідними формулами наведено у підрозділі 2.3.

2.2 Алгоритм побудови RAG-систем

У роботі запропоновано алгоритм для RAG-систем на основі Contextual Retrieval, адаптований під аналіз фінансових звітів. Ключова відмінність від стандартного підходу: фрагмент іде в індекс не сам по собі, а вже збагачений макро-контекстом усього документа. Зв'язок між локальним шматком і його глобальною рамкою закладається одразу у векторі – а не добудовується потім через ре-ранжування.

Алгоритм містить три модифікації стандартного конвеєра:

– інтелектуальний парсинг через Docling: замість витягування «голого» тексту система задіює модель Granite-Docling (VLM), яка розпізнає структуру документа через DocTags. Результат – ієрархія «заголовок-текст» зберігається, таблиці залишаються цілими, і числові показники не втрачають прив'язку до своїх заголовків уже на етапі введення даних [29];

– метод послідовної контекстуалізації: Кожен фрагмент c_i не просто векторизується, а проходить через етап генерації ситуативного префікса P_i . Математично здійснюється перехід від функції векторизації $E(c_i)$ до

функції E($P_i \oplus c_i$), де $\oplus$ – операція текстової конкатенації. Префікс P_i синтезується моделлю gpt-5-nano за формулою:

$$P_i = LLM(c_i, M, H_{window}), \quad (2.3)$$

де $P_i -$ ситуативний префікс, що синтезується для контекстуалізації окремого інформаційного вузла;

c_i – макро-контекст, який містить повний текст документа;

M – глобальний контекст або метадані документа;

H_{window} – параметричне вікно або буфер описів.

– алгоритм ковзного вікна для збереження змісту: Для подолання проблеми обмеженого контекстного вікна та оптимізації витрат впроваджено буфер описів розміром k=3. Це забезпечує "семантичний шлейф", який дозволяє моделі розв'язувати анафори ("цей показник", "за згаданий період") ще на етапі створення індексу. Формалізацію механізму контекстуалізації з обробкою граничних випадків подано на рисунку 2.2.

Алгоритм усуває семантичну амнезію напряму: глобальний контекст документа прив'язується до кожного фрагмента ще на етапі індексації – не після пошуку, а до нього. Кожна згенерована цифра простежується до першоджерела. Метрики точності на фінансових задачах це підтверджують.

Технологія Docling, розроблена IBM Research, є інтелектуальним інструментом для конвертації документів, який виходить за межі можливостей традиційного OCR [29]. На відміну від класичних рішень, що розглядають PDF як набір символів з координатами, Docling використовує уніфіковану візуально-мовну модель (VLM), таку як Granite-Docling (SmolDocling), що інтегрує розуміння макета та тексту в єдиний компактний архітурний блок [29].

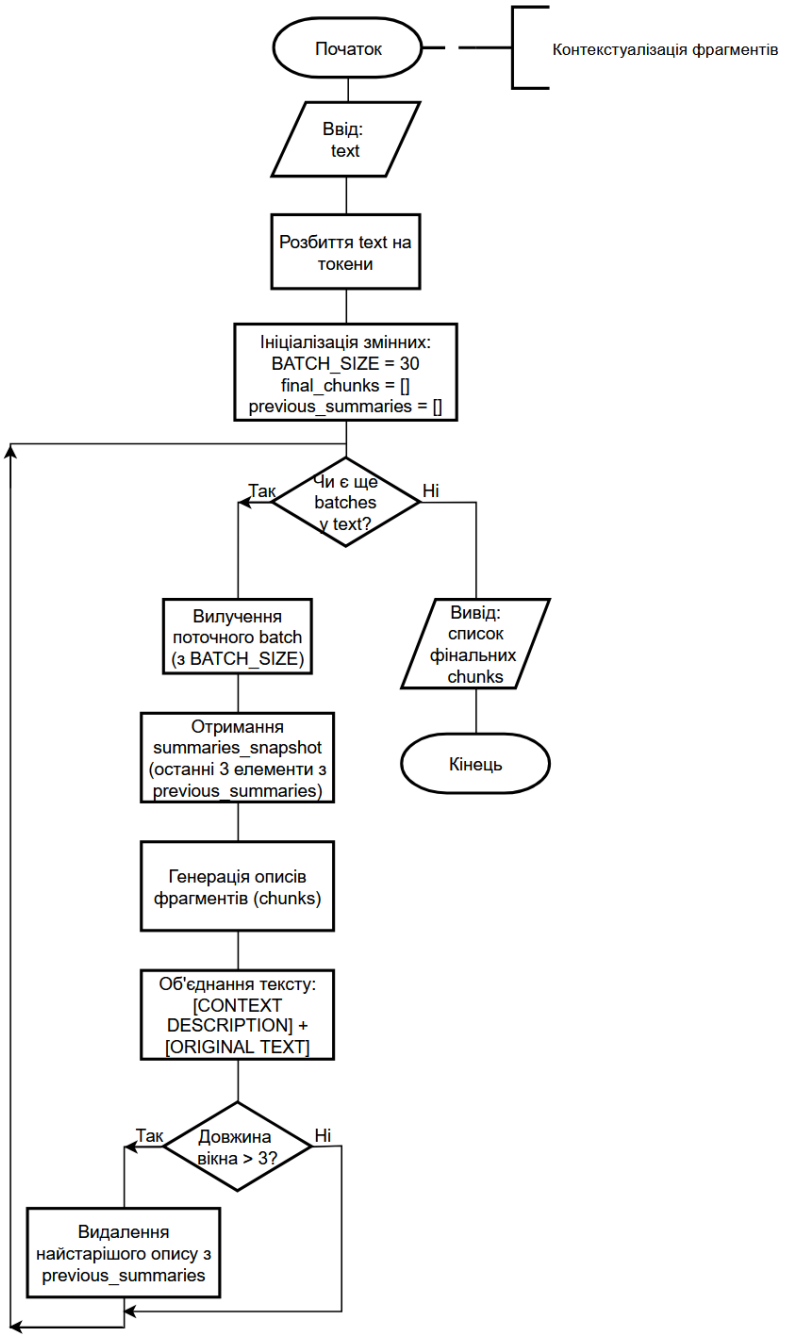

Рисунок 2.2 – Блок-схема контекстуалізації з ковзним вікном

Основним викликом при обробці PDF є те, що цей формат був розроблений для візуальної ідентичності, а не для збереження логічної структури. Традиційні парсери часто стикаються з ефектом «word salad», коли текст із різних колонок або таблиць змішується в один незв’язний потік. Docling вирішує цю задачу за допомогою багатостадійного конвеєра [29]:

– аналіз макета. Моделі глибокого навчання (Heron-101, RT-DETRv2), навчені на корпусі DocLayNet, виявляють заголовки, параграфи, таблиці, рисунки й підписи як окремі об'єкти з bounding boxes;

– реконструкція ієрархії заголовків. Аналіз розміру шрифту, накреслення та нумерації відновлює дерево документа з рівнями H1–H6;

– прив'язка підписів. Алгоритми просторової близькості групують captions із відповідними рисунками й таблицями, що згодом дає LLM змогу «читати» вміст рисунка через його підпис на етапі пошуку [29];

– використання DocTags: формат розмітки DocTags знає, що є заголовком, що таблицею, а що – службовим колонтитулом. Токенів витрачається менше, а контент і структура не змішуються в одну кашу [29].

У сфері RAG-систем вибір інструменту для індексації визначає якість всієї системи. Порівняння Docling з такими бібліотеками, як PyMuPDF чи Unstructured, виявляє суттєві розбіжності в підходах і візуально показано в таблиці 2.2.

Таблиця 2.2 – Порівняльний аналіз інструментів парсингу документів

Характеристика	Docling	PyMuPDF	Unstructured
Основний механізм	VLM + Layout Analysis	PDF-координати	Ансамбль моделей + OCR
Робота з таблицями	TableFormer (97.9% точність)	Базове вилучення тексту	OCR-орієнтоване (75%)
Ієрархія заголовків	Реконструює за DocTags	Лише за наявності метаданих	Часткове розпізнавання
Швидкість	1.3–6.8 стор/сек	Дуже висока	Низька (залежить від OCR)
Вихідний формат	Markdown, JSON, DocTags	Markdown, Text	Markdown, JSON, HTML

Перевага Docling полягає у використанні формату Markdown для представлення таблиць. Markdown зберігає структуру таблиці у вигляді текстових координат, що дозволяє LLM сприймати взаємозв’язок «рядок-стовпець» без складного парсингу HTML-тегів. Це дозволяє зберігати метадані таблиць (заголовки колонок) безпосередньо в кожному фрагменті тексту при chunking [29].

Загальна логіка взаємодії компонентів системи, від моменту отримання запиту користувача до формування кінцевої відповіді, представлена в додатку А на рисунку А.1, а також на рисунку А.2. При цьому в додатку А перші рисунки

відповідають базовій схемі роботи Naive RAG, а наступні – вдосконаленій схемі роботи Advanced RAG, що є предметом даного дослідження.

2.3 Обґрунтування метрик оцінювання та вибору архітектурних рішень

Доцільність використання запропонованого алгоритму Advanced RAG підтверджується порівняльним аналізом на еталонному наборі даних FinanceBench за допомогою фреймворку RAGAS. Вибір RAGAS як метрологічного базису зумовлений його здатністю декомпозувати помилки системи на рівень ретривера та рівень генератора, використовуючи парадигму «LLM-as-a-judge». Критерії ефективності, за якими проводилося обґрунтування:

1) Answer Correctness (Правильність відповіді): Визначає ступінь відповідності згенерованої відповіді еталонній (Ground Truth) [30]. Вона розраховується як зважене середнє між фактичною точністю (на основі порівняння тверджень) та семантичною подібністю. Розрахунок виконується за формулою:

$$\text{Answer Correctness} = \omega_1 * FS + \omega_2 * SS, \tag{2.4}$$

де ω_1, ω_2 – вагові коефіцієнти (за замовчуванням у Ragas використовуються ваги 0.75 для фактів та 0.25 для семантики);

SS – семантична подібність, що обчислюється як косинусна відстань між векторними представленнями (embeddings) відповіді та еталона;

FS – фактична подібність, що базується на аналізі окремих тверджень (claims) через F1-міру:

$$FS = \frac{|TP|}{|TP| + 0.5(|FP| + |FN|)} \tag{2.5}$$

де TP – факти, що присутні і в еталоні, і у відповіді;

FP – факти, що присутні у відповіді, але відсутні в еталоні (галюцинації);

FN – факти, що є в еталоні, але модель їх пропустила.

На відміну від класичних лінгвістичних метрик (BLEU, ROUGE), які оцінюють лише синтаксичний збіг токенів, поєднання SS та FS дозволяє враховувати як смислове наповнення, так і строгу фактологічну точність. Використання цієї метрики у межах тестування на еталонному наборі даних FinanceBench уможливлює математичне підтвердження того, що синергія запропонованого ретривера та збагаченого контексту знижує рівень помилок генератора та максимізує точність відповідей у межах фінансового аналізу.

2) Context Recall (Повнота вилучення контексту): Здатність системи знаходити всі факти, необхідні для відповіді [30]. Вона оцінює, наскільки повно вилучений контекст покриває ключові твердження еталонної відповіді. Метрика розраховується за формулою:

$$Context\ Recall = \frac{|C_{attr}|}{|C_{total}|} \tag{2.6}$$

де C_{attr} (Attributed Claims) – кількість окремих тверджень (факторів) в еталонній відповіді, які безпосередньо підтверджуються інформацією з вилучених фрагментів контексту;

C_{total} (Total Claims) – загальна кількість фактичних тверджень, що містяться в ідеальній (еталонній) відповіді.

3) Context Precision (Точність вилучення контексту): Оцінює якість ранжування вилучених фрагментів (чанків) [30]. Ця метрика визначає, наскільки ефективно система ставить релевантні документи на перші позиції у видачі. Розрахунок виконується за формулою:

$$Context\ Precision = \frac{\sum_{k=1}^{n}(P_k * v_k)}{R} \tag{2.7}$$

де n – загальна кількість чанків, які система повернула за один запит;

k – порядковий номер (ранг) фрагмента у видачі (від 1 до n);

v_k – індикатор релевантності k-го фрагмента (двійкове значення: 1, якщо фрагмент корисний для відповіді; 0, якщо фрагмент не релевантний).

R – загальна кількість релевантних фрагментів, що потрапили у видачу (фактично це сума всіх v_k);

P_k – точність на рівні k-го фрагмента.

P_k показує частку релевантних документів серед усіх документів, що знаходяться на позиціях від 1 до k і розраховується як:

$$P_k = \frac{TP_k}{k} \tag{2.8}$$

де TP_k – кількість «істинно позитивних» (релевантних) фрагментів серед перших k позицій.

4) Faithfulness (Фактологічна достовірність): Відсутність галюцинацій та заземлення відповіді виключно на знайдених даних [30]. Вона оцінює, чи кожне твердження у згенерованому тексті логічно випливає з наданого контексту. Метрика розраховується за формулою:

$$\text{Faithfulness} = \frac{\text{Supported claims}}{\text{Total claims}} \tag{2.9}$$

де Supported claims – кількість окремих тверджень у відповіді, які безпосередньо підтверджуються (верифікуються) інформацією з вилученого контексту;

Total claims – загальна кількість фактичних тверджень, на які була розбита згенерована відповідь для перевірки.

Вибір моделі gpt-5-nano для генерації описів обґрунтований її унікальною економічною характеристикою: 5 центів за мільйон токенів при збереженні високої швидкодії (TTFT 0.93s). Це робить запропонований метод Contextual Retrieval

промислово масштабованим, оскільки вартість повної індексації мільйонів документів стає доступною для корпоративного сектору. Математичне моделювання показує, що такий підхід у 2026 році є більш вигідним, ніж використання моделей з надвеликим контекстним вікном (Gemini 2.0/Claude 4), де вартість запиту зростає лінійно від довжини документа, а затримки (latency) стають критичними для UX.

Застосування Qdrant як векторної бази даних аргументовано підтримкою Filterable HNSW, що дозволяє виконувати логічну ізоляцію даних департаментів у багатокористувацьких середовищах без втрати продуктивності пошуку. Також можливість використання квантування (Scalar/Binary) дозволяє зменшити потреби в RAM у 4-32 рази, що знижує TCO системи.

Обмеження методу та шляхи їх подолання:

– latency при індексації: Велика кількість викликів API для генерації описів уповільнює наповнення бази. Подолано шляхом впровадження асинхронних батчів по 30 чанків та семафорів;

– вартість токенів: Повторна передача документа для кожного чанка. Подолано через Prompt Caching, що дає 90% знижку на зчитування вже завантаженого контенту;

– шум у префіксах: Можливі помилки LLM при синтезі контексту. Подолано через жорсткий формат виводу (Structured JSON) та системну валідацію схем.

У підсумку, запропонований комплекс методів та обраних моделей дозволяє створити систему семантичного пошуку, яка задовольняє найсуворішим вимогам точності та прозорості корпоративних інтелектуальних систем 2026 року.

3 РЕАЛІЗАЦІЯ АЛГОРИТМІВ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

3.1 Опис набору даних та умови проведення експериментальних досліджень

Якість RAG-системи у фінансовому домені починається не з архітектури – а з даних і того, як вони структуровані на вході. Для експерименту обрано FinanceBench: перший відкритий benchmark, де LLM перевіряють саме на реальній фінансовій звітності. Не на синтетичних питаннях і не на вікіпедії – на SEC-файлингах, де є ієрархічні таблиці, перехресні посилання між розділами і питання, що вимагають числового логічного виведення. Це і є робочий день фінансового аналітика.

Основою набору даних є офіційні документи, що подаються компаніями до Комісії з цінних паперів і бірж США (SEC). Ці документи характеризуються надзвичайною щільністю фактичної інформації та специфічною термінологією. Структура корпусу документів FinanceBench включає:

– річні звіти за формою 10-K, що містять детальні фінансові таблиці, обговорення результатів діяльності керівництвом (MD&A), фактори ризику та примітки до консолідованої звітності. Середній обсяг такого документа становить близько 150 – 250 сторінок, що створює значне навантаження на контекстне вікно моделі;

– квартальні звіти за формою 10-Q, які забезпечують оновлення фінансових показників протягом року та часто містять порівняльні таблиці за аналогічні періоди попередніх років;

– поточні звіти за формою 8-K, що фіксують суттєві корпоративні події, такі як злиття, поглинання або зміни в раді директорів;

– релізи про доходи та транскрипти конференц-дзвінків, де неструктурований текст поєднується з числовими прогнозами.

Загальна кількість верифікованих експертами пар «питання-відповідь» у наборі перевищує 10 000, проте для даного дослідження використано вибірку зі 150 складних багатокрокових запитів. Ці запити розподілені за категоріями складності: 28% стосуються прямого вилучення фактів, 66% вимагають числових розрахунків (наприклад, обчислення коефіцієнтів ліквідності або темпів зростання) і 6%

потребують логічного синтезу інформації з різних частин документа. У таблиці 3.1 представлено детальні статистичні характеристики використаного набору даних.

Таблиця 3.1 – Статистичний огляд набору даних FinanceBench

Показник	Значення	Опис
Загальна кількість документів	84 SEC filings	Основний корпус для вибірки 150 питань
Типи документів	10-K, 10-Q, 8-K, Earnings	Річна, квартальна та оперативна звітність
Кількість тестових запитів	150	Експертно анотована вибірка
Середня довжина 10-K	> 60 000 токенів	Еквівалент 150+ сторінок тексту
Розподіл за GICS секторами	9 із 11 класів	Включаючи IT, BFSI, охорону здоров'я та ін.
Модальність даних	Текст + Таблиці	Висока частка структурованої цифрової інформації

Обидві проблеми – «семантична амнезія» і «невидимий займенник» – мають одне джерело. Naive RAG нарізає документи на шматки фіксованого розміру, по 512 токенів, без урахування змісту. Фрагмент отримує цифру – «чистий прибуток $15 млрд» – але назва компанії і фінансовий рік залишаються в іншому шматку. Вектор такого фрагмента в латентному просторі неповний: за запитом «прибуток компанії X за 2023 рік» він просто не спливе.

Для вирішення цієї проблеми підготовка даних у розробленій системі базується на використанні бібліотеки Docling (IBM Research). Docling забезпечує інтелектуальний парсинг документів, який виходить за межі простого вилучення тексту. Архітектура Docling інтегрує моделі глибокого навчання для аналізу макета (layout analysis) та розпізнавання структур таблиць. Процес підготовки включав наступні кроки:

– перетворення PDF у структурований Markdown: модель Heron-101 (VLM) розпізнає заголовки, підзаголовки, списки і примітки, зберігаючи ієрархію документа. У фінансових звітах структура – це частина змісту: показник у розділі «Активи» і той самий показник у розділі «Зобов'язання» – це різні речі. Якщо ієрархія губиться на вході, жодне збагачення потім не допоможе;

– реконструкція таблиць. Замість того, щоб зчитувати таблиці як послідовність рядків, Docling використовує модель TableFormer для відновлення зв'язків між комірками [29]. Отримані таблиці конвертуються у формат Markdown, що дозволяє моделі векторизації сприймати структуру даних, а не просто набір цифр;

– семантична розмітка через DocTags. Система додає метаінформацію до кожного елемента, що дозволяє в подальшому відокремлювати основний зміст від колонтитулів та службової інформації, яка створює шум у векторному просторі.

Для порівняльного аналізу було реалізовано дві стратегії фрагментації. У Naive RAG застосовувався стандартний RecursiveCharacterTextSplitter із розміром чанка 512 токенів та перекриттям 15%. У вдосконаленій архітектурі Advanced RAG кожен фрагмент, отриманий після інтелектуального парсингу, проходив етап контекстного збагачення (Contextual Retrieval). Це передбачало генерацію ситуативного префікса, який жорстко пов'язує знання про весь документ до кожного окремого вузла інформації. Математично цей процес препроцесингу спрямований на те, щоб векторне представлення фрагмента в латентному просторі було максимально повним та однозначним.

Парсинг запускався асинхронно – без черг, батчами. На GPU-інфраструктурі це дало від 1.90 до 6.82 сторінок за секунду; весь корпус FinanceBench підготували за кілька годин. Результат цього етапу – не просто розпарсений текст. Кожен фрагмент виходить із нього вже з контекстом: він знає, з якого документа він, з якого розділу і що було до нього.

3.2 Реалізація та налаштування алгоритмів

Центральною частиною дослідження є програмна реалізація архітектури Advanced RAG, яка інтегрує вдосконалені алгоритми контекстного пошуку, семантичної векторизації та динамічного управління індексами у векторній базі даних Qdrant. Основна інновація полягає у переході від пасивного вилучення інформації до активного збагачення даних на етапі індексації (Input & Enrichment phase).

Для подолання деградації контексту при фрагментації реалізовано метод Contextual Retrieval, запропонований фахівцями Anthropic, але адаптований для використання з новою моделлю gpt-5-nano. Модель gpt-5-nano була обрана як основний контекстуалізатор завдяки її унікальним характеристикам, що робить технологію економічно доступною для промислових масштабів 2026 року.

Програмна логіка модуля збагачення реалізує алгоритм «ковзного вікна» (sliding window) для збереження смислової неперервності документа. Кожен текстовий фрагмент c_i перед векторизацією проходить через LLM, яка генерує для нього стислий ситуативний префікс P_i.

Програмну реалізацію алгоритму послідовної контекстуалізації з буфером попередніх описів подано у додатку Б (лістинг Б.1). Асинхронний конвеєр обробляє пакети по 30 фрагментів одночасно через asyncio.gather, а буфер previous_summaries_window забезпечує семантичний шлейф із трьох останніх описів.

Буфер попередніх описів вирішує саме цю задачу. Якщо в поточному фрагменті написано «ці витрати» – gpt-5-nano бачить три попередні описи і розуміє: R&D компанії NVIDIA, Q2 2024. Анафора знімається ще на етапі індексації, не в рантаймі. Системний промпт жорстко фіксував формат виводу: Entity Period: Topic – і нічого зайвого. Це не естетика, а практика: щільний, однозначний префікс прямо впливає на те, наскільки точно вектор потрапить у потрібну латентну околицю.

Програмну реалізацію модуля синтезу описів, який звертається до моделі gpt-5-nano, наведено у додатку Б (лістинг Б.2). Системний промпт жорстко фіксує

структуру заголовка та забороняє повторення тематик із попередніх описів, що мінімізує семантичний шум у векторному просторі.

Для реалізації цього процесу було розроблено асинхронний конвеєр на Python з використанням бібліотеки asyncio. Обробка документів відбувалася пакетами (batches) по 30 фрагментів одночасно, що дозволило максимально завантажити API провайдера без перевищення лімітів (RPM/TPM). Для мінімізації витрат застосовано механізм Prompt Caching: оскільки для кожного чанка одного документа передається один і той самий «макро-текст», OpenAI API кешує його, надаючи 90% знижку на вхідні токени.

Векторна база даних Qdrant була налаштована на роботу з векторами розмірністю d=3072. Ключовим етапом оптимізації стало налаштування алгоритму HNSW для забезпечення високої повноти (recall) при мінімальній затримці. Параметри індексації були обрані на основі аналізу щільності векторного простору фінансових документів. Конкретні значення гіперпараметрів HNSW у Qdrant подано в таблиці 3.2. Вибір зміщений у бік повноти (recall) за рахунок часу індексації, оскільки у фінансовому аудиті пропущений релевантний фрагмент коштує дорожче за зайві мілісекунди затримки.

Таблиця 3.2 – Гіперпараметри HNSW-індексу в Qdrant

Параметр	Значення	Призначення та обґрунтування вибору
m	32	Кількість двонаправлених зв'язків на вузол на кожному шарі
ef_construct	256	Розмір динамічного списку кандидатів при побудові графа.
hnsw_ef (runtime)	128	Розмір списку кандидатів на запиті.
full_scan_threshold	10 000	Нижче цього порога Qdrant обходить HNSW і виконує brute-force.
on_disk	false	Індекс тримається в RAM повністю. При d=3072 один вектор займає 12,3 КБ (float32); 50K векторів ≈ 600 МБ – прийнятно для тестового стенду.

Python 3.11, Qdrant, LangChain, gpt-5-nano – кожен компонент закриває конкретну задачу в конвеєрі. Разом вони роблять одне: система не шукає схожі слова – вона знає, про кого і про який період йдеться. У фінансовому аудиті та юридичному консалтингу це не перевага над конкурентами.

3.3 Результати експериментів та їх аналіз

Експериментальна валідація запропонованої архітектури Advanced RAG проводилася на наборі даних FinanceBench з використанням фреймворку RAGAS. Фреймворк RAGAS був обраний як метрологічний базис через його здатність об'єктивно оцінювати якість конвеєра в розрізі окремих компонентів: ретривера (вилучення) та генератора (синтез відповіді).

У процесі тестування було проведено порівняння базової системи (Naive RAG) та розробленої системи (Advanced RAG) на складних фінансових запитах. Отримані результати продемонстрували статистично значуще покращення за всіма ключовими метриками (Таблиця 3.3).

Таблиця 3.3 – Порівняльна оцінка ефективності на наборі FinanceBench

Категорія метрики	Naïve RAG	Advanced RAG	Відносне покращення
Answer Correctness	0.3063	0.3584	+17.00%
Faithfulness	0.6137	0.6621	+7.88%
Context Recall	0.2067	0.2800	+35.46%
Context Precision	0.2803	0.3868	+37.99%

Аналіз отриманих даних дозволяє зробити наступні висновки щодо впливу впроваджених алгоритмів на якість системи:

– Context Recall [30]. Зростання цієї метрики на 35.46% є прямим наслідком впровадження методу Contextual Retrieval та інтелектуального парсингу через Docling. У Naive RAG механічне розбиття фінансових таблиць призводило до того, що релевантні показники (наприклад, чистий дохід) вилучалися без контексту емітента та року. Математична формула по якій розраховувалась ця метрика

підтверджує, що вдосконалена система знаходить на 35% більше критично важливих фактів, необхідних для відповіді. Практичним підтвердженням цього є сценарій, зображений на рисунку 3.1: система Naive RAG на запит щодо причин зростання витрат SG&A повернула відповідь «I can't find the answer», хоча відповідна інформація була присутня в корпусі. Провал retrieval стався через те, що деталі судових витрат і виходу з ринку були розміщені в середині секції MD&A і після механічного розбиття потрапили до ізольованого фрагмента без тематичного заголовка.

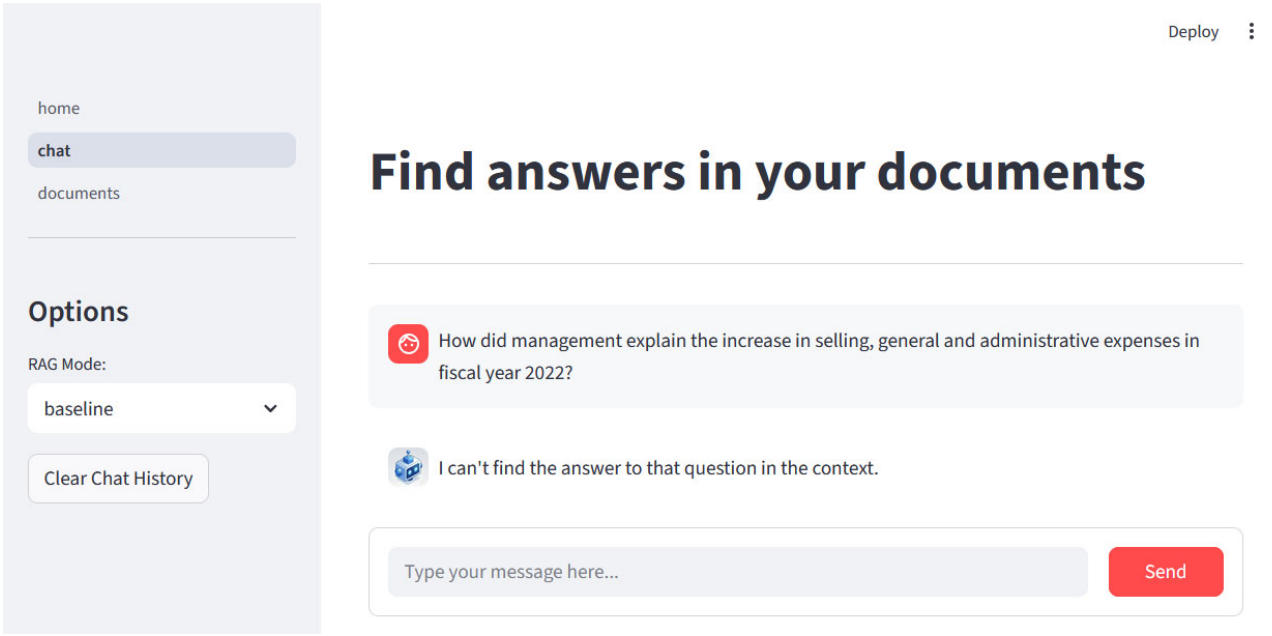

Рисунок 3.1 – Відповідь системи Naive RAG: відмова від відповіді через втрату контекстного зв'язку

Система Advanced RAG, як показано на рисунку 3.2, успішно відтворила повний контекст із секції MD&A: судовий позов Combat Arms на суму $1.2B, витрати, пов'язані зі сполуками PFAS, та одноразові витрати на вихід з російського ринку. Кожен із цих фрагментів у векторній базі мав у префіксі ідентифікатор розділу та назву емітента, що дозволило ретриверу знайти всі релевантні вузли та передати їх генератору.

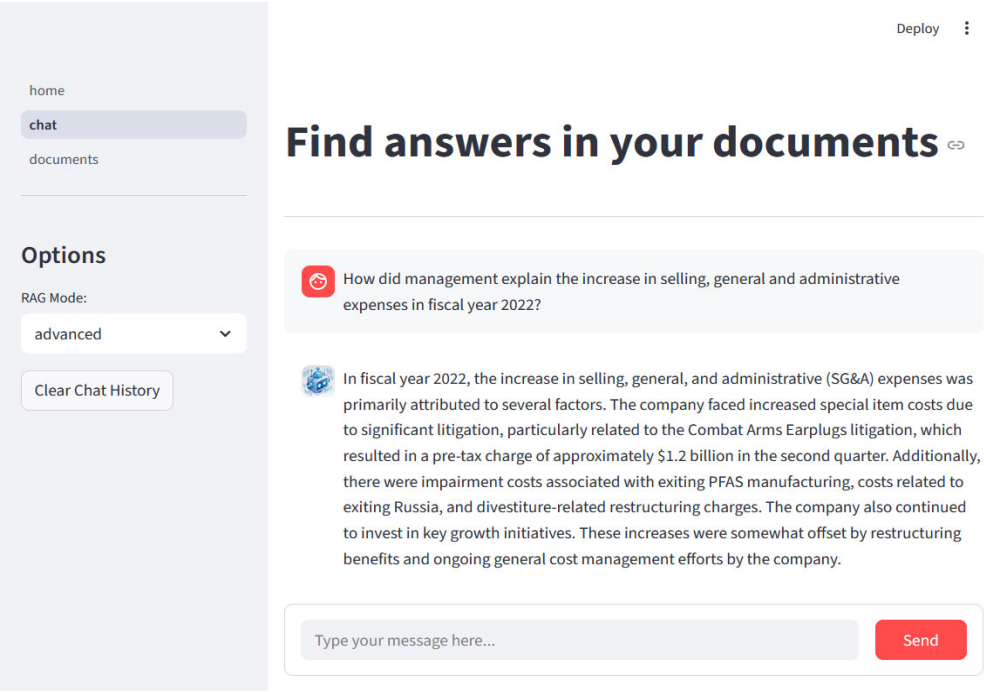

Рисунок 3.2 – Відповідь системи Advanced RAG: повна деталізована відповідь із контексту MD&A

– Context Precision [30]. Покращення на 37.99% свідчить про те, що додавання ситуативних префіксів gpt-5-nano дозволило математично згрупувати вектори у відповідні латентні околиці, мінімізуючи «лексичні колізії». Система стала значно краще ранжувати релевантні фрагменти, виводячи їх у топ-видачу (Top-K), що критично для моделей із обмеженим контекстним вікном.

– Faithfulness [30]. Показник зріс на 7.88%, що вказує на суттєве зниження частоти фактологічних «галюцинацій». Оскільки генератор отримує контекст, де кожна цифра інтегрована до суб'єкта та періоду, він перестає конфабулювати відсутні зв'язки. Це забезпечує жорстку залежність згенерованої відповіді від знайдених даних. Для ілюстрації цього ефекту на рисунку 3.3 наведено відповідь системи Naive RAG на запит щодо причин падіння чистого прибутку компанії. Система впевнено сформулювала відповідь, проте змішала дані кількох різних емітентів – згадала роздрібний сектор, COVID-19 та показник $0.02/share, що не відповідає жодному документу з корпусу. Це є класичним прикладом фактологічної «галюцинації», яка виникає через відсутність чіткої прив'язки фрагменту до конкретної компанії та звітного періоду.

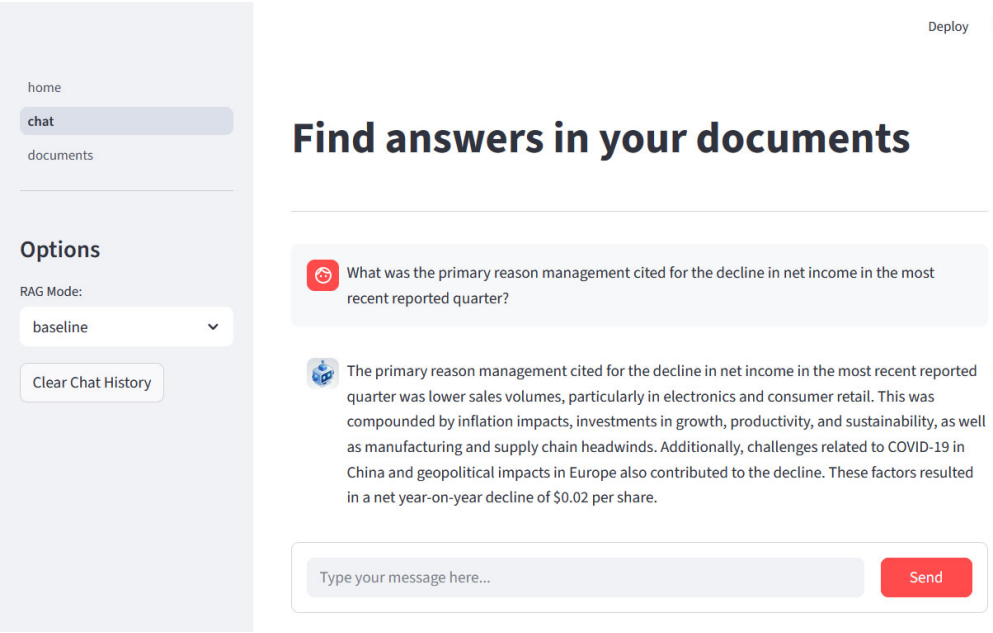

Рисунок 3.3 – Відповідь системи Naive RAG: галюцинація через відсутність контекстної прив'язки

На рисунку 3.4 представлено відповідь системи Advanced RAG на ідентичний запит. Завдяки ситуативному префіксу, який містить назву емітента та звітний рік, генератор точно ідентифікував механізм падіння – зростання provision for credit losses на 29% у річному вимірі до рівня $1.8B – та сформував верифіковану відповідь без конфабуляції.

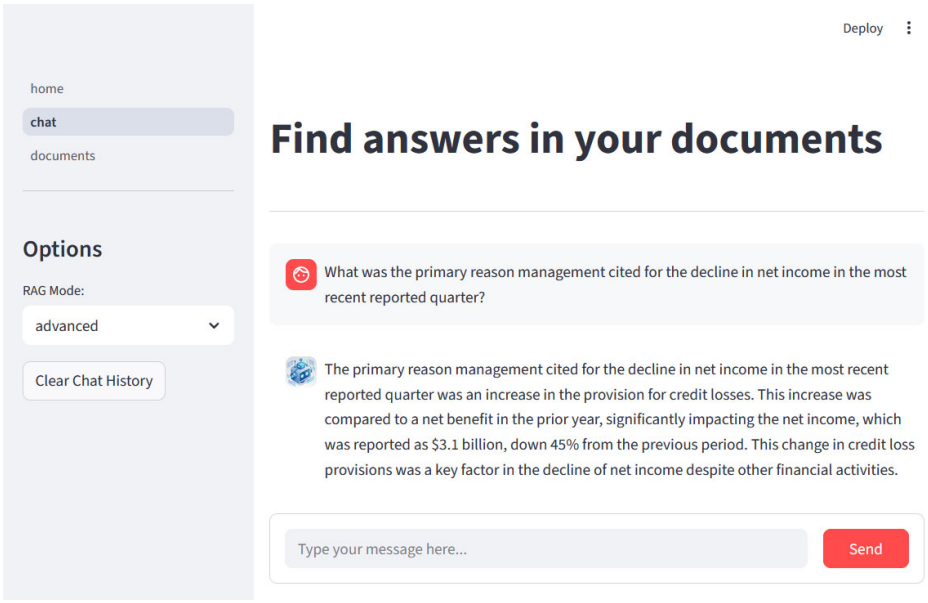

Рисунок 3.4 – Відповідь системи Advanced RAG: точна ідентифікація причини на основі заземленого контексту

– Answer Correctness [30]. Загальне підвищення точності на 17% підтверджує доцільність використання запропонованого підходу для створення надійних інтелектуальних асистентів корпоративного рівня. На відміну від суто семантичних метрик, Answer Correctness враховує як фактичну точність (F1-score), так і семантичну подібність до еталонної відповіді експерта. Показовим прикладом є запит щодо динаміки gross profit margin за FY2022. На рисунку 3.5 видно, що Naive RAG повернув значення 39.6% – 39.0% та зробив висновок про зростання за рахунок channel mix – відповідь, яка є правдоподібною, проте стосується іншого емітента. Система не мала можливості верифікувати приналежність числових даних, оскільки векторне представлення фрагмента не містило назви компанії.

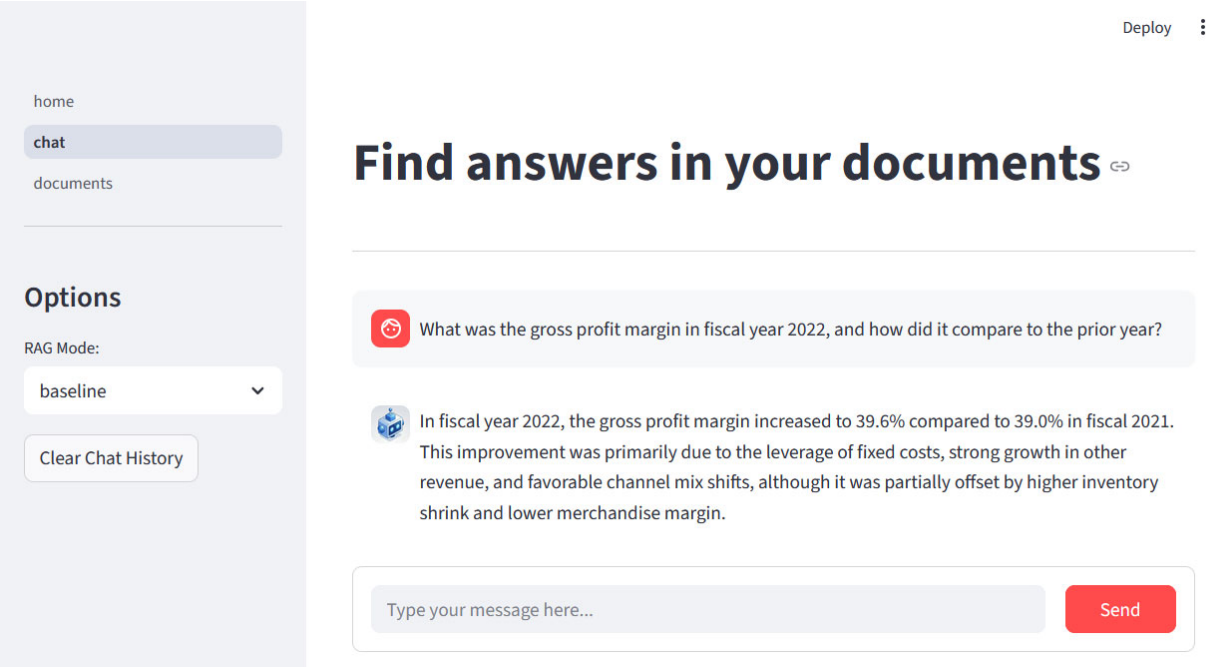

Рисунок 3.5 – Відповідь системи Naive RAG: правдоподібна, але хибна відповідь через підміну компанії

Advanced RAG, як відображено на рисунку 3.6, коректно ідентифікував емітента як AMD та надав верифіковані значення: валова маржа зросла з 45% до 48% у зв'язку зі збільшенням частки придбаної компанії Xilinx у загальному портфелі та одночасним зростанням амортизаційних нарахувань. Відповідь є трейсованою – кожне числове значення прив'язане до конкретного документа в індексі.

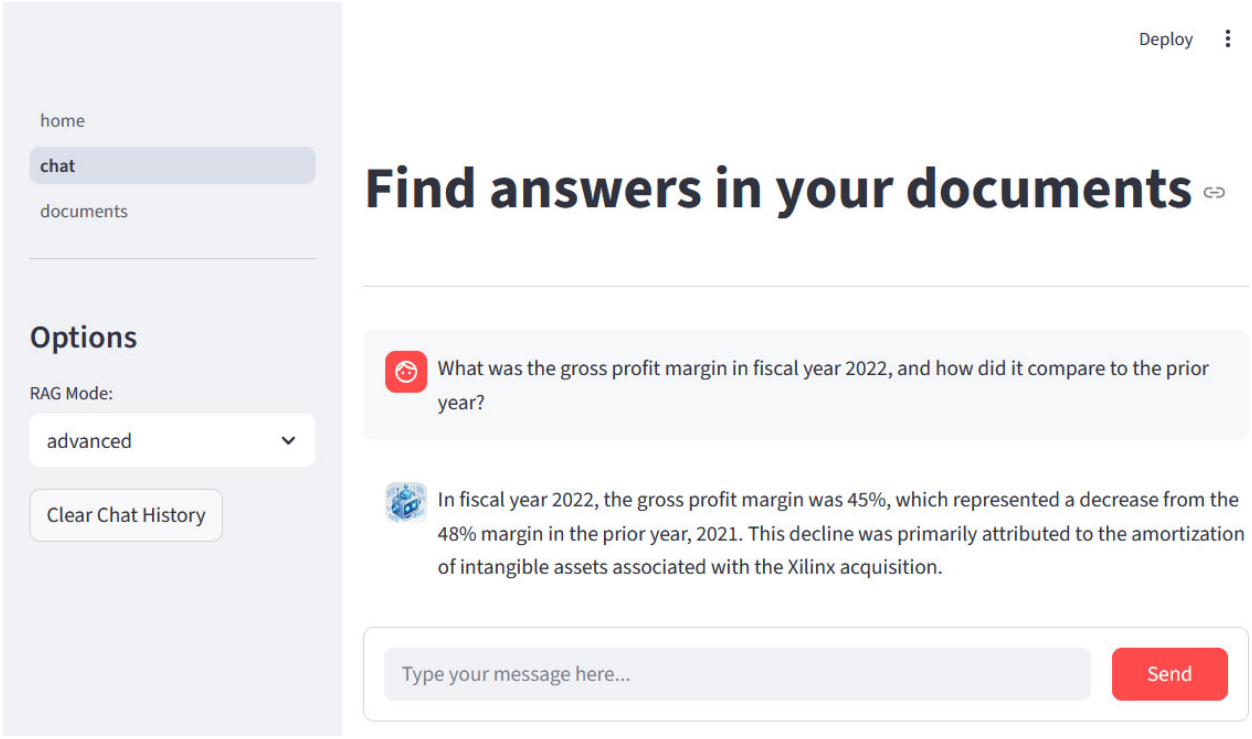

Рисунок 3.6 – Відповідь системи Advanced RAG: верифікована відповідь із правильним емітентом та джерелом

Результати виявили один нетривіальний патерн: якість вилучення контексту і точність числових розрахунків пов'язані напряму. Quick Ratio вимагає одночасно Total Inventories і Total Current Liabilities. Naive RAG розкидав їх по різних чанках – ретривер знаходив одне, пропускав інше, формула давала збій. Advanced RAG отримував обидві змінні разом, бо макро-контекст у кожному фрагменті утримував їхній зв'язок. Генератор не став розумнішим – він просто нарешті отримав усі вхідні дані.

Експеримент спростував тезу про те, що моделі з надвеликим контекстним вікном (наприклад, Gemini 2.0 Pro з 2M токенами) роблять RAG непотрібним. Хоча такі моделі здатні прийняти весь звіт 10-K цілком, вони страждають від феномену «Lost in the Middle» – зниження точності вилучення фактів, що знаходяться в середині довгого промпту. Впроваджений метод Contextual Retrieval забезпечив вищу точність заземлення при значно менших витратах часу та коштів.

Збагачення фрагментів на індексації коштує ресурсів. Проте gpt-5-nano коштує п'ять центів за мільйон токенів, а Prompt Caching кешує текст документа

між запитами – тобто кожен наступний фрагмент одного PDF передається зі знижкою 90%. Обробка 1000 документів по 50 сторінок вийшла приблизно в $90.

Затримка є лише при побудові індексу – це одноразово. Запит користувача в рантаймі HNSW у Qdrant обробляє за мілісекунди незалежно від розміру бази.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи досягнуто поставленої мети – розроблено вдосконалений алгоритм RAG-системи на основі методу Contextual Retrieval, що мінімізує ризик фактологічних галюцинацій та підвищує точність вилучення даних зі складних корпоративних документів у сферах фінансів та юридичного консалтингу.

1. Проаналізовано стан ринку корпоративних великих мовних моделей у секторах Banking, Financial Services і Insurance (BFSI) та юридичного консалтингу, а також досліджено природу фактологічних «галюцинацій» та еволюцію RAG-архітектур, таких як Naive RAG, Contextual Retrieval та Agentic RAG. Порівняльний аналіз існуючих підходів дозволив ідентифікувати ключову складність механічної фрагментації документів, відомою як ефект «семантичної амнезії» та анафоричні розриви між суміжними сегментами, що унеможливлюють точне вилучення корпоративних даних. Проведений аналіз дозволив зробити постановку задач щодо застосування методів контекстного збагачення фрагментів на етапі індексації.

2. Розроблено метод послідовної контекстуалізації фрагментів тексту на основі концепції Contextual Retrieval із використанням моделі gpt-5-nano для синтезу ситуативних макро-префіксів. На відміну від існуючих підходів, що залучають великі мовні моделі, кожному чанку генерується стислий префікс, який прив'язує його до глобального змісту документа; застосування gpt-5-nano в поєднанні з механізмом Prompt Caching робить метод економічно доступним для промислових баз знань. Результатом є усунення ефекту «семантичної амнезії» без експоненційного зростання витрат на контекстуалізацію.

3. Реалізовано алгоритм «ковзного вікна» із динамічним буфером k=3 попередніх описів, що дозволив забезпечити семантичну зв'язність між суміжними сегментами документа та усунути анафоричні посилання на етапі побудови векторного індексу.

4. Проведено структурний аналіз документів засобами бібліотеки Docling зі збереженням ієрархічної організації заголовків і таблиць, що дозволило обгрунтувати вибір компонентів технологічного стека (Qdrant, text-embedding-3-

large, LangChain) відповідно до вимог продуктивності, масштабованості та економічної доцільності промислового застосування.

5. Сформовано єдиний асинхронний конвеєр процедури індексації корпоративних документів, що становить цілісну програмну основу архітектури Advanced RAG. Експериментальна валідація розробленої архітектури Advanced RAG відносно базової Naive RAG на спеціалізованому наборі даних FinanceBench із використанням метрик фреймворку RAGAS засвідчила перевагу запропонованої архітектури: повнота вилучення контексту зросла на 35,46%, точність ранжування – на 37,99%, фактологічна достовірність – на 7,88%, правильність відповідей – на 17,00%.

6. Верифіковано гіпотезу про перевагу методу контекстного збагачення фрагментів над збільшенням контекстного вікна LLM, за рахунок чого встановлено, що запропонований підхід забезпечує вищу точність вилучення даних за менших обчислювальних витрат. Отримані результати спростовують тезу про надлишковість RAG-архітектур в умовах застосування LLM із розширеним контекстним вікном.

7. Практичним результатом дослідження є програмний модуль із високим ступенем готовності до промислового впровадження в інтелектуальних системах семантичного пошуку (сектори BFSI та юридичний консалтинг). Стабільну швидкодію у процесі масштабування бази знань забезпечує алгоритм HNSW, а повна простежуваність кожної згенерованої відповіді до першоджерела відповідає суворим вимогам фінансового аудиту.

8. Сформульовано рекомендації для подальших досліджень, які передбачають перехід до архітектури Agentic RAG з автономним ітеративним уточненням запитів, а також інтеграцію протоколу MCP для забезпечення прямого доступу до динамічних корпоративних джерел даних.

9. Основні результати дослідження висвітлено у науковій публікації: Коваль В. С., Крушельницький С. М. «Підхід до побудови RAG-систем на основі удосконаленого методу контекстного пошуку», опублікованій у науковому фаховому виданні категорії «Б» – електронному журналі «Наука і техніка сьогодні».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Аналіз обсягу, частки та галузі ринку корпоративних LLM. URL: https://www.snsinsider.com/reports/enterprise-llm-market-9102 (дата звернення 02.06.2026).
2. Велика мовна модель. Вікіпедія, Вільна енциклопедія. 2024. URL: https://en.wikipedia.org/wiki/Large_language_model (дата звернення 02.06.2026).
3. Гао Й. та ін. Пошукова доповнена генерація для великих мовних моделей: огляд. arXiv preprint. 2023. URL: https://arxiv.org/pdf/2312.10997 (дата звернення 02.06.2026).
4. Документація метрик. RAGAS. 2024. URL: https://docs.ragas.io/en/latest/concepts/metrics/index.html (дата звернення 02.06.2026).
5. Звіт про дослідження та статистику галюцинацій ШІ 2026. URL: https://suprmind.ai/hub/insights/ai-hallucination-statistics-research-report-2026/ (дата звернення 02.06.2026).
6. Іслам П. та ін. FinanceBench: новий еталон для відповідей на фінансові запитання. arXiv preprint. 2023. URL: https://arxiv.org/pdf/2311.11944v1 (дата звернення 02.06.2026).
7. Казларіс І., Антоніу Е., Діамантарас К., Братсас К. Від ілюзії до розуміння: таксономічний огляд методів пом'якшення галюцинацій у LLM. AI. 2025. Вип. 6(10). С. 260. DOI: https://doi.org/10.3390/ai6100260 (дата звернення 02.06.2026).
8. Контекстний пошук. Новини Anthropic. 2024. URL: https://www.anthropic.com/news/contextual-retrieval (дата звернення 02.06.2026).
9. Косинусна подібність. Вікіпедія, Вільна енциклопедія. 2024. URL: https://en.wikipedia.org/wiki/Cosine_similarity (дата звернення 02.06.2026).
10. Льюїс П. та ін. Пошукова доповнена генерація для складних завдань NLP. Advances in Neural Information Processing Systems. 2020. Т. 33. С. 9459-9474. URL: https://arxiv.org/pdf/2005.11401 (дата звернення 02.06.2026).
11. Махарджан А., Ядав У. Чанкінг, пошук та переранжування: емпірична оцінка архітектур RAG для відповідей на запитання за програмними документами. arXiv preprint. 2026. URL: https://arxiv.org/html/2601.15457v1 (дата звернення 02.06.2026).

12. Метрики оцінювання RAG. Блог Confident AI. 2024. URL: https://www.confident-ai.com/blog/rag-evaluation-metrics (дата звернення 02.06.2026).
13. Набір даних PatronusAI/financebench. Hugging Face. 2023. URL: https://huggingface.co/datasets/PatronusAI/financebench (дата звернення 02.06.2026).
14. Найкращі стратегії чанкінгу для RAG у 2025 році. Firecrawl. 2025. URL: https://www.firecrawl.dev/blog/best-chunking-strategies-rag-2025 (дата звернення 02.06.2026).
15. Обмеження чанкінгу та пошуку в QA. Reddit (r/Rag). 2024. URL: https://www.reddit.com/r/Rag/comments/1jh2xgs/limitations_of_chunking_and_retrieval_in_qa (дата звернення 02.06.2026).
16. Одхіамбо Т. Найефективніший підхід до RAG на сьогодні: контекстний пошук Anthropic та гібридний пошук. Medium. 2024. URL: https://medium.com/@tom.odhiambo/the-most-effective-rag-approach-to-date-anthropics-contextual-retrieval-and-hybrid-search (дата звернення 02.06.2026).
17. Парадигма чанкінгу: рекурсивна семантика для оптимізації RAG. Матеріали 8-ї Міжнародної конференції з обробки природної мови та мовлення (ICNLSP-2025). 2025. URL: https://aclanthology.org/2025.icnlsp-1.15.pdf (дата звернення 02.06.2026).
18. Посібник з контекстуальних ембедінгів (Contextual Embeddings Guide). URL: https://platform.claude.com/cookbook/capabilities-contextual-embeddings-guide (дата звернення 02.06.2026).
19. Про схвалення Концепції розвитку штучного інтелекту в Україні : Розпорядження Кабінету Міністрів України від 2 грудня 2020 р. № 1556-р. 2020. URL: https://zakon.rada.gov.ua/laws/show/1556-2020-p (дата звернення 02.06.2026).
20. Сансев'єро О. Ембединги речень. Блог Hackerllama. 2024. URL: https://osanseviero.github.io/hackerllama/blog/posts/sentence_embeddings2/ (дата звернення 02.06.2026).
21. Статистика галюцинацій LLM. URL: https://sqmagazine.co.uk/llm-hallucination-statistics/ (дата звернення 02.06.2026)..

22. Хуанг Л. та ін. Огляд галюцинацій у великих мовних моделях: принципи, таксономія, виклики та відкриті питання. arXiv preprint. 2023. URL: https://arxiv.org/html/2311.05232v2 (дата звернення 02.06.2026).
23. Agentic RAG: архітектурні патерни, які дійсно працюють для корпоративного ШІ. URL: https://dedicatted.com/insights/agentic-rag-architecture-patterns-that-actually-work-for-enterprise-ai (дата звернення 02.06.2026).
24. Agnihotri P. Building Agentic Adaptive RAG with LangGraph for Production. Artificial Intelligence in Plain English. 2025. URL: https://ai.plainenglish.io/building-agentic-rag-with-langgraph-mastering-adaptive-rag-for-production-c2c4578c836a (дата звернення 02.06.2026).
25. Contextual retrieval в Anthropic з використанням баз знань Amazon Bedrock. URL: https://aws.amazon.com/blogs/machine-learning/contextual-retrieval-in-anthropic-using-amazon-bedrock-knowledge-bases/ (дата звернення 02.06.2026).
26. Enterprise LLM Market Size, Share & Industry Analysis. URL: https://www.fortunebusinessinsights.com/enterprise-llm-market-114178 (дата звернення 02.06.2026).
27. FinanceBench: тести для оцінки фінансових LLM. URL: https://www.emergentmind.com/topics/financebench (дата звернення 02.06.2026).
28. Liu N. F. et al. Lost in the Middle. 2023. URL: https://arxiv.org/pdf/2307.03172 (дата звернення 02.06.2026)..
29. Livathinos N. та ін. Docling: An Efficient Open-Source Toolkit for AI-driven Document Conversion. arXiv preprint. 2025. URL: https://arxiv.org/pdf/2501.17887 (дата звернення 02.06.2026).
30. Metrics. Ragas. 2023. URL: https://docs.ragas.io/en/v0.1.21/concepts/metrics/index.html (дата звернення 02.06.2026).
31. Model Context Protocol (MCP). URL: https://www.oracle.com/database/model-context-protocol-mcp/ (дата звернення 02.06.2026).
32. Vaswani A. та ін. Attention Is All You Need. Advances in Neural Information Processing Systems. 2017. URL: https://arxiv.org/pdf/1706.03762 (дата звернення 02.06.2026).

33. Qdrant – векторна база даних для пошуку на основі штучного інтелекту. Офіційна документація. URL: https://qdrant.tech/documentation/ (дата звернення 02.06.2026).

34. LangChain. Документація фреймворку для розробки застосунків на основі LLM. URL: https://python.langchain.com/docs/introduction/ (дата звернення 02.06.2026).

35. Robertson S., Zaragoza H. The Probabilistic Relevance Framework: BM25 and Beyond. Foundations and Trends in Information Retrieval. 2009. Vol. 3(4). P. 333–389. URL: https://www.staff.city.ac.uk/~sbrp622/papers/foundations_bm25_review.pdf (дата звернення 02.06.2026).

36. Коваль В. С., Крушельницький С. М. Підхід до побудови RAG-систем на основі удосконаленого методу контекстного пошуку. Наука і техніка сьогодні. 2026. № 4 (58). С. 3470–3481. DOI: https://doi.org/10.52058/2786-6025-2026-4(58)-3470-3481.

37. Комар М. П., Саченко А. О., Васильків Н. М., Гладій Г. М., Коваль В. С., Ліп'яніна-Гончаренко Х. В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Штучний інтелект» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 57 с.

Додаток А

Схема роботи RAG-систем

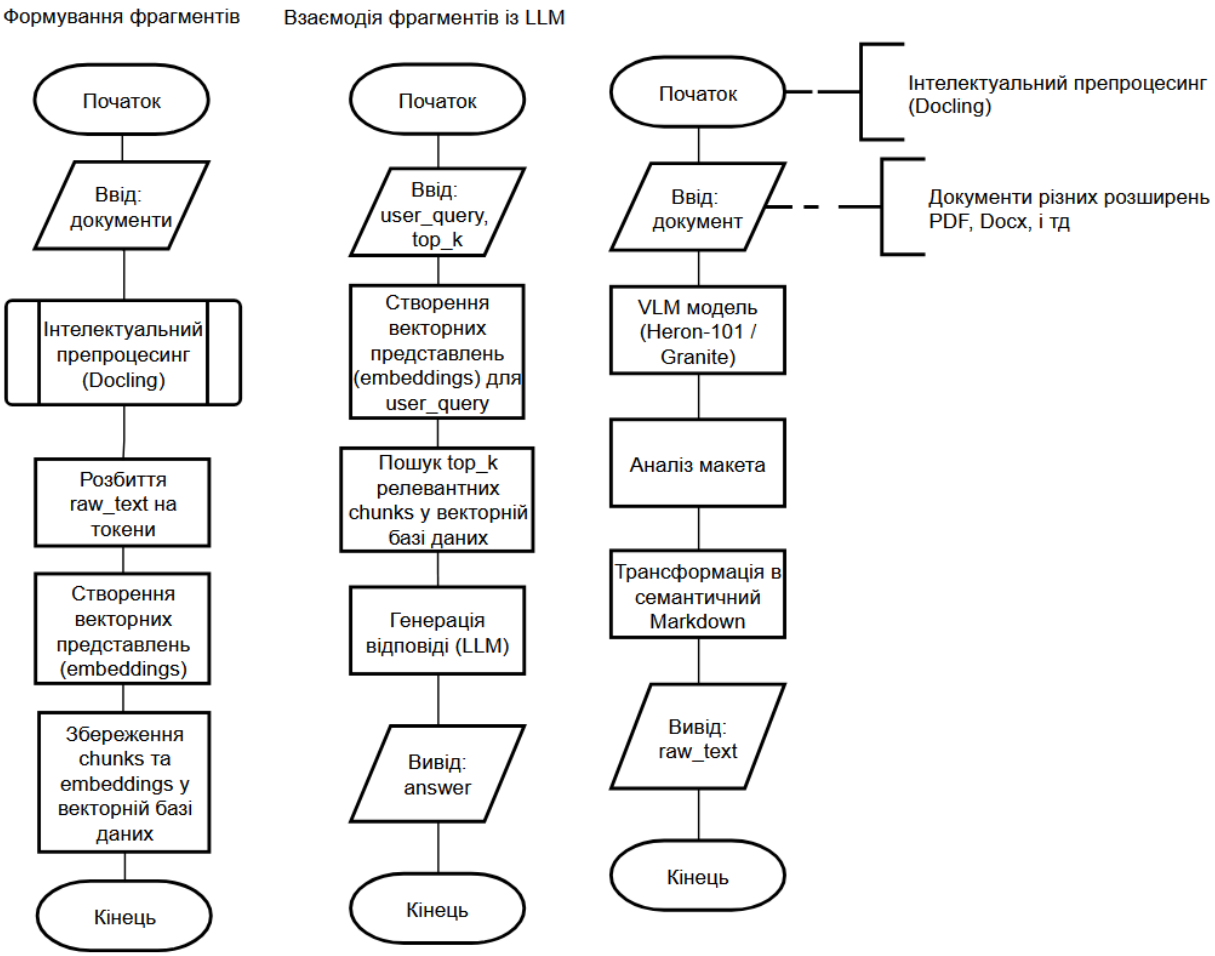

Рисунок А.1 – Схема роботи алгоритму Naive RAG

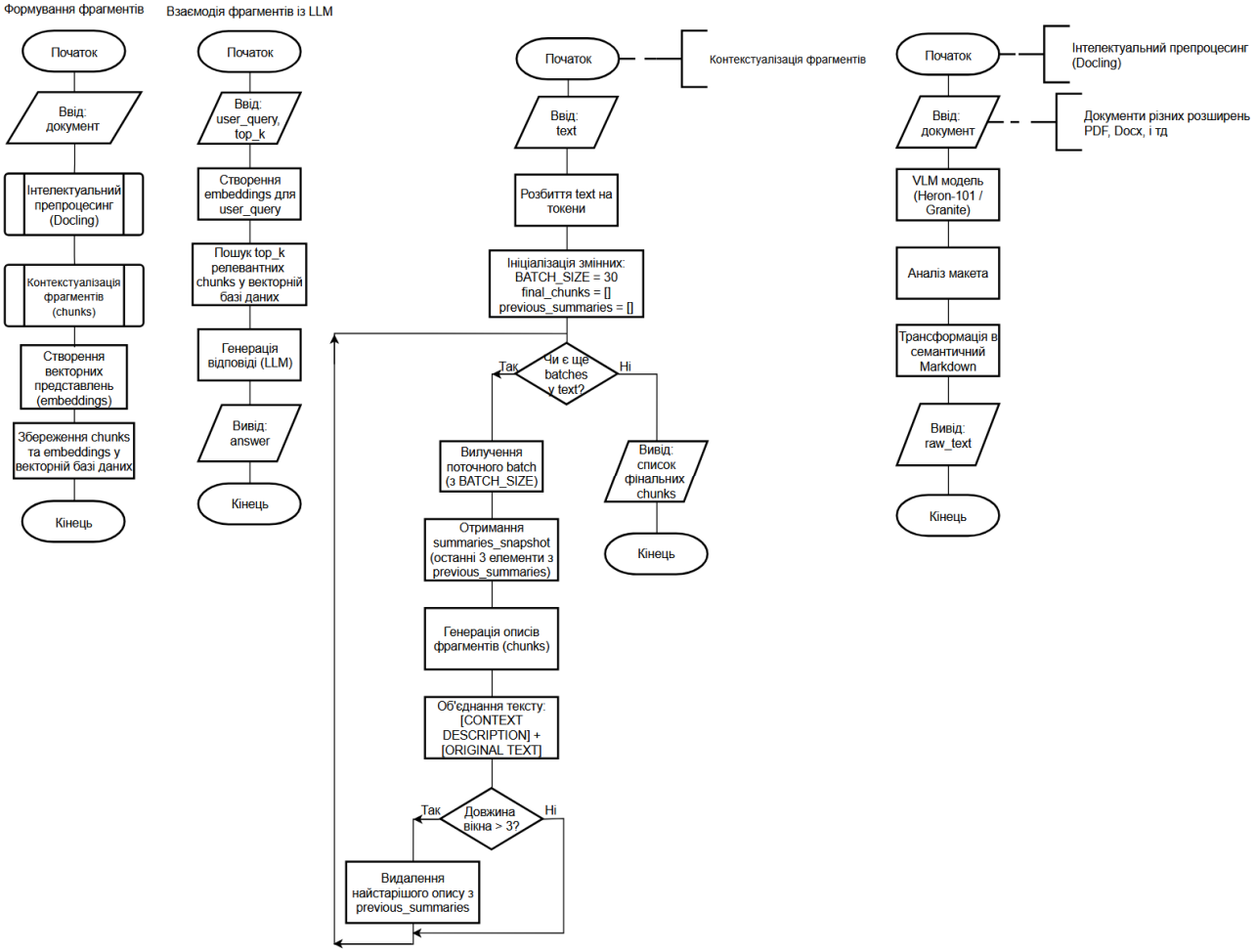

Рисунок А.2 – Схема роботи алгоритму Advanced RAG

Додаток Б

Програмний код реалізації підсистеми контекстного збагачення

Лістинг Б.1 – Алгоритм послідовної контекстуалізації з ковзним вікном(модуль chunker.py):

```
async def process_advanced(
    self, text_to_split: str, filename: str
) -> list[Chunk]:
    """Послідовна контекстуалізація фрагментів з ковзним вікном (k=3)."""
    split_text: list[str] = self.__split_text_baseline(text_to_split)
    final_chunks: list[Chunk] = [None] * len(split_text)
    BATCH_SIZE = 30
    previous_summaries_window: list[str] = []
    for batch_start in range(0, len(split_text), BATCH_SIZE):
        batch = split_text[batch_start : batch_start + BATCH_SIZE]
        summaries_snapshot = list(previous_summaries_window[-3:])

        tasks = [
            self.llm_handler.generate_chunk_description(
                current_chunk_text=content,
                previous_summaries=summaries_snapshot,
                filename=filename,
                max_concurrent_requests=BATCH_SIZE,
            )
            for content in batch
        ]
        descriptions = await asyncio.gather(*tasks, return_exceptions=False)
        for i, (content, description) in enumerate(zip(batch, descriptions)):
            chunk_order = batch_start + i
            enriched_content = (
                f"[CONTEXT DESCRIPTION]:\n{description}\n\n"
```

```
            f"[ORIGINAL TEXT]:\n{content}"
        )
        final_chunks[chunk_order] = Chunk(
            content=enriched_content,
            metadata=Metadata(
                filename=filename,
                order_number_of_chunk=chunk_order,
            ),
        )
        previous_summaries_window.append(description)
        if len(previous_summaries_window) > 3:
            previous_summaries_window.pop(0)

    return final_chunks
```

Лістинг Б.2 – Синтез ситуативного префікса через gpt-5-nano (модуль openai_handler.py):

```
async def generate_chunk_description(
    self,
    current_chunk_text: str,
    previous_summaries: list[str],
    filename: str,
    max_concurrent_requests: int = 30,
) -> str:
    context_history = (
        "\n".join(f"- {s}" for s in previous_summaries)
        if previous_summaries
        else "No previous context (Start of document)."
    )

    system_prompt = """You are an expert financial document summarizer
```

```
creating headers for RAG indexing.

FORMAT: [Type] Entity Period: Specific Topics
RULES:
1. Type: [Announcement], [Table], [CEO Statement], [MD&A], [Risk], etc.
2. Entity & Period: extract from previous summaries if present,
   else from chunk. Format: "Company FY2022" or "Company Q3 2023".
3. Specific Topics (2-4 categories): prefer metric names over vague terms;
   never repeat topics already listed in previous headers.
4. Length: 10-12 words total maximum.
"""
    user_prompt = f"""CONTEXT:
Previous summaries (DO NOT repeat their topics):
{context_history}

CURRENT CHUNK TO SUMMARIZE:
{current_chunk_text}

Generate header:
"""
    semaphore = asyncio.Semaphore(max_concurrent_requests)
    async with semaphore:
        response = await self.async_client.chat.completions.create(
            model="gpt-5-nano",
            messages=[
                {"role": "system", "content": system_prompt},
                {"role": "user", "content": user_prompt},
            ],
        )
        return response.choices[0].message.content.strip()
```

Додаток В

Апробація отриманих результатів

Видавнича група «Наукові перспективи»

Всеукраїнська Асамблея докторів наук із державного управління

«Наука і техніка сьогодні»

Випуск № 4(58) 2026

Київ – 2026

ISSN 2786-6025 Online

УДК 001.32:1 /3](477)(02)

R40-05553

DOI: Crossref we use DOIs **https://doi.org/10.52058/2786-6025-2026-4(58)**

«Наука і техніка сьогодні» (Серія «Педагогіка», Серія «Право», Серія «Економіка», Серія «Фізико-математичні науки», Серія «Техніка»)»: журнал. 2026. № 4(58) 2026. С. 5059

Згідно наказу Міністерства освіти і науки України від 07.04.2022 № 320 журналу присвоєно категорію "Б" із економіки та педагогіки (спеціальності – 015 - Педагогічні науки; 076 - Економічні науки)

Згідно наказу Міністерства освіти і науки України від 06.06.2022 № 530 журналу присвоєно категорію "Б" із права (спеціальність – 081 Юридичні науки)

Згідно наказу Міністерства освіти і науки України від 10.10.2022 № 894 журналу присвоєно категорію "Б" із техніки (спеціальність - 122 Комп'ютерні науки)

Журнал видається за підтримки Міждержавної гільдії інженерів консультантів, Інституту філософії та соціології Національної Академії Наук Азербайджану (Баку, Азербайджан), громадської організації «Християнська академія педагогічних наук України» та громадської організації «Всеукраїнська асоціація педагогів і психологів з духовно-морального виховання»

Рекомендовано до видавництва Президією Всеукраїнської Асамблеї докторів наук з державного управління (Рішення від 24.04.2026, № 8/4-26)

Google INDEX COPERNICUS Academic Resource Index ResearchBib Журнал включено до міжнародної наукометричної бази Index Copernicus (IC), міжнародної пошукової системи Google Scholar та до міжнародної наукометричної бази даних Research Bible

Згідно Порядку формування Переліку наукових фахових видань України, затвердженого наказом МОН України від 15.01.2018 № 32, повнотекстовий доступ до наукових статей журналу представлений на платформі «Наукова періодика України» в Національній бібліотеці України імені В.І. Вернадського НАН України та в Національному репозитарії академічних текстів

Головний редактор:

Коренева Інна Миколаївна *- доктор педагогічних наук, професор, декан факультету природничої і фізико-математичної освіти Глухівського національного педагогічного університету імені Олександра Довженка; професор кафедри теорії і методики викладання природничих дисциплін Глухівського національного педагогічного університету імені Олександра Довженка (Україна)*

Редакційна колегія:

№ 4(58) 2026
НАУКА і ТЕХНІКА СЬОГОДНІ
ISSN 2786-6025 Online

№ 4(58) 2026
НАУКА і ТЕХНІКА СЬОГОДНІ
ISSN 2786-6025 Online
УДК 004.8, 681.3

https://doi.org/10.52058/2786-6025-2026-4(58)-3470-3481

Коваль Василь Сергійович кандидат технічних наук, доцент, заступник декана факультету комп'ютерних інформаційних технологій, доцент кафедри інформаційно-обчислювальних систем і управління, Західноукраїнський національний університет, м. Тернопіль, https://orcid.org/0000-0003-4726-097X

Крушельницький Святослав Мар'янович студент 4-го курсу факультету комп'ютерних інформаційних технологій, Західноукраїнський національний університет, м. Тернопіль, https://orcid.org/0009-0001-1257-9051

ПІДХІД ДО ПОБУДОВИ RAG–СИСТЕМ НА ОСНОВІ УДОСКОНАЛЕНОГО МЕТОДУ КОНТЕКСТНОГО ПОШУКУ

Анотація. Розвиток великих мовних моделей (LLM) забезпечив автоматизацію інтелектуальних завдань, проте проблема фактологічних «галюцинацій» залишається критичним бар'єром для їхнього впровадження у високоточних доменах, таких як фінанси та право. Стандартна архітектура доповненої пошукової генерації (Retrieval-Augmented Generation, RAG) вирішує проблему актуальності знань, але стикається з деградацією контексту через фрагментацію даних. У статті представлено результати дослідження та розробки вдосконаленого підходу до побудови RAG-систем на основі методу контекстного пошуку (Contextual Retrieval). Основна увага приділена вирішенню проблеми розв'язання анафори і збереженню семантичної цілісності через програмне збагачення ізольованих фрагментів тексту глобальним ситуативним макроконтекстом. Запропоновано архітектуру, що використовує модель gpt-5-nano для синтезу описів та метод «ковзного вікна» для оптимізації обчислювальних витрат. Експериментальна оцінка на базі набору даних FinanceBench за допомогою метрик RAGAS продемонструвала суттєве покращення показників повноти та точності вилучення контексту, що прямо корелює зі зниженням рівня фактологічних помилок у згенерованих відповідях. Станом на 2026 рік, коли системи штучного інтелекту перейшли від експериментальних прототипів до критично важливих середовищ виконання знань, такий підхід стає галузевим стандартом для забезпечення економічної безпеки. Використання моделі gpt-5-nano, дозволяє інтегрувати потужні когнітивні можливості при наднизькій вартості всього п'ять центів за мільйон токенів, що робить технологію контекстного збагачення економічно доступною для масового впровадження. Впроваджений алгоритм ковзного вікна підтримує динамічну пам'ять про

ISSN 2786-6025 Online

попередні фрагменти документа, фактично усуваючи ефект семантичної амнезії, коли цифрові показники втрачають зв'язок із конкретними суб'єктами або часовими рамками через механічне розбиття тексту. Результати тестування зафіксували зростання повноти контексту на тридцять п'ять відсотків та точності на тридцять вісім відсотків, що є критичним для автоматизації фінансового аудиту та юридичного консалтингу. Пропонована методологія забезпечує повну трасованість кожної генерації до першоджерела. Робота доводить, що контекстне збагачення є більш ефективною стратегією, ніж просте використання моделей із надвеликим контекстним вікном, оскільки забезпечує вищу точність при значно менших обчислювальних витратах у промислових масштабах 2026 року.

Ключові слова: штучний інтелект, великі мовні моделі, RAG, пошукова генерація, контекстний пошук, Contextual Retrieval, обробка природної мови, NLP, фінансовий аналіз.

Koval Vasyl Serhijoovych Doctor of Philosophy , Associate Professor, Vice-Dean of the Faculty of Computer Information Technologies, Associate Professor of the Department of Information and Computing Systems and Control, Western Ukrainian National University, Ternopil, https://orcid.org/0000-0003-4726-097X

Krushelnytskyi Sviatoslav Maryanovych 4th-year Student, Faculty of Computer Information Technologies, Western Ukrainian National University, Ternopil, https://orcid.org/0009-0001-1257-9051

AN APPROACH TO BUILDING RAG SYSTEMS BASED ON THE ENHANCED CONTEXTUAL RETRIEVAL METHOD

Abstract. The development of large language models (LLMs) has enabled the automation of intellectual tasks; however, the problem of factual "hallucinations" remains a critical barrier to their adoption in high-stakes domains such as finance and law. The standard Retrieval-Augmented Generation (RAG) architecture addresses the issue of knowledge relevance but suffers from context degradation due to data fragmentation. This paper presents the results of research and development of an improved approach to building RAG systems based on the Contextual Retrieval method. The main focus is on resolving anaphora and preserving semantic integrity through the programmatic enrichment of isolated text fragments with a global situational macro-context. An architecture is proposed that uses the gpt-5-nano model for description synthesis and the "sliding window" method for optimizing computational costs. Experimental evaluation based on the FinanceBench dataset using RAGAS metrics demonstrated a significant improvement in the completeness

ISSN 2786-6025 Online

and accuracy of context extraction, which directly correlates with a reduction in the level of factual errors in the generated responses. As of 2026, when artificial intelligence systems have transitioned from experimental prototypes to mission-critical knowledge execution environments, this approach has become the industry standard for ensuring economic security. The use of the gpt-5-nano model allows for the integration of powerful cognitive capabilities at an ultra-low cost of just five cents per million tokens, making contextual enrichment technology economically accessible for mass adoption. The implemented sliding-window algorithm maintains a dynamic memory of previous document segments, effectively eliminating the effect of semantic amnesia, where digital metrics lose their connection to specific subjects or time frames due to mechanical text segmentation. Test results recorded a 35% increase in contextual completeness and a 38% increase in accuracy, which is critical for the automation of financial auditing and legal consulting. The proposed methodology ensures full traceability of each generation back to the original source. The work demonstrates that contextual enrichment is a more effective strategy than simply using models with an oversized context window, as it provides higher accuracy at significantly lower computational costs on an industrial scale in 2026.

Keywords: artificial intelligence, large language models, RAG, search-assisted generation, contextual search, Contextual Retrieval, natural language processing, NLP, financial analysis.

Постановка проблеми. Сучасний етап розвитку глобальної цифрової економіки у 2024 – 2026 роках характеризується інтенсивним впровадженням систем штучного інтелекту в процеси обробки неструктурованої інформації. Великі мовні моделі (LLM), такі як GPT-4, Claude 3 та Gemini, змінили парадигму взаємодії людини з даними, продемонструвавши здатність до кодування, логічного виведення та синтезу текстів на рівні, що наближається до людського. Проте широка експлуатація цих моделей у корпоративному секторі, де точність є критичною вимогою, стримується критичною вразливістю – схильністю до «галюцинацій» [1, 3]. Оскільки параметрична пам'ять моделей є статичною і обмеженою даними моменту завершення навчання, вони не мають доступу до динамічної внутрішньої документації підприємств, що призводить до генерування правдоподібних, але фактологічно хибних відповідей [13].

В контексті України, де «Концепція розвитку штучного інтелекту» визначає пріоритетним вдосконалення систем публічного управління та економічної безпеки, розробка надійних методів застосування ШІ на перевірені джерела знань стає стратегічним завданням [30]. Архітектура пошукової генерації (RAG) виникла як відповідь на цей виклик, дозволяючи моделям вилучати актуальні факти з зовнішніх векторних баз даних перед формуванням

ISSN 2786-6025 Online

відповіді [5]. Ця технологія розділяє когнітивні функції системи на параметричні знання (ваги нейронної мережі) та непараметричні знання (зовнішні індекси), що теоретично має гарантувати стовідсоткову трасованість кожного твердження [5, 13]. Проте практичне впровадження базових RAG-систем (Naive RAG) у таких складних доменах, як фінансовий аудит та юридичний консалтинг, виявило фундаментальний недолік: процес фрагментації тексту руйнує його семантичну цілісність [4, 7]. Механічне розбиття багатосторінкових звітів на дрібні сегменти фіксованого розміру призводить до втрати контекстуальних зв'язків, особливо коли ключові сутності або часові рамки відокремлені від числових показників [8, 12]. Це породжує феномен «семантичної амнезії», коли вилучений фрагмент містить цифру, але не «пам'ятає», до якої компанії чи року вона відноситься. Таким чином, розробка методів контекстного збагачення даних на етапі індексації є критично важливим кроком для створення безпечних та ефективних інтелектуальних систем корпоративного рівня.

Аналіз останніх досліджень і публікацій. Фундаментальні засади архітектури пошукової генерації (RAG) були формалізовані у 2020 році групою дослідників під керівництвом П. Льюїса [5], які запропонували поєднання щільного векторного пошуку з генеративними трансформерами. Ця парадигма дозволила розділити пам'ять системи на параметричну (ваги моделі) та непараметричну (зовнішній індекс), забезпечуючи «заземлення» відповідей у фактичних даних.

Станом на початок 2026 року фокус наукового пошуку змістився з простої реалізації конвеєра на якість представлення та структурування інформації. У закордонній науковій спільноті домінуючим напрямом став перехід від «Naive RAG» до Advanced RAG, що спричинило наступні покращення [7, 13]:

- контекстне збагачення (Contextual Retrieval): фахівці компанії Anthropic обґрунтували необхідність відмови від обробки ізольованих фрагментів на користь їхнього ситуативного збагачення макро-префіксами, що розв'язало проблему «невидимого займенника» та семантичної фрагментації, що раніше призводило до втрати логічних зв'язків;
- гібридні архітектури та ранжування: дослідники з Google та OpenAI інтегрують алгоритми гібридного пошуку (семантичні вектори + лексичний пошук BM25) та крос-енкодери для точного переранжування (Reranking) результатів [16, 26];
- агентні системи (Agentic RAG): новітній тренд 2025–2026 років направлений на використання автономних агентів, які здатні розбивати запити на підзадачі та взаємодіяти з даними через протоколи на кшталт MCP (Model Context Protocol) [11, 26].

ISSN 2786-6025 Online

Попри появу моделей із надвеликим контекстним вікном (Gemini 1.5/2.0 Pro), дослідження 2025 року підтверджують, що RAG залишається більш економічно вигідним та масштабованим рішенням для корпоративних баз знань обсягом у терабайти [11, 13]. Це зумовлено експоненційним зростанням вартості обчислень та затримками (latency) при завантаженні повних документів [13, 26]. Нижче наведено порівняння характеристик, що обґрунтовує переваги інтегрованого підходу (таблиця 1):

Таблиця 1

Порівняння характеристик [5, 13]

Характеристика	**Параметрична пам'ять (LLM)**	**Непараметрична пам'ять (RAG індекс)**
Джерело знань	Внутрішні ваги після навчання	Зовнішні бази даних, PDF, API
Актуальність	Статична (дата навчання)	Динамічна (моментальне оновлення)
Масштабованість	Обмежена вікном контексту	Майже необмежена (терабайти)
Точність фактів	Схильність до галюцинацій	Висока (прямі цитати)
Трасованість	Відсутня	Повна (посилання на джерела)

Мета статті – створення вдосконаленого підходу до побудови RAG-систем на основі методу контекстного пошуку (Contextual Retrieval) для мінімізації ризику фактологічних галюцинацій та підвищення точності вилучення даних зі складних корпоративних документів. Досягнення поставленої мети передбачає розроблення алгоритму методу та його інтеграцію в програмне забезпечення. Впровадження ізольованих фрагментів тексту із ситуаційним макро-контекстом та використанням моделі gpt-5-nano дозволить здійснити експериментальну перевірку ефективності запропонованого рішення на спеціалізованому наборі даних за допомогою метрик фреймворку RAGAS.

Виклад основного матеріалу. Класична архітектура Naive RAG базується на лінійному процесі «Retrieve-then-Generate», де ключовим етапом підготовки даних є механічна сегментація тексту за допомогою інструментів на кшталт RecursiveCharacterTextSplitter. Такий підхід передбачає розбиття

ISSN 2786-6025 Online

документа на фрагменти фіксованого розміру за кількістю символів або токенів, що часто ігнорує семантичну структуру та логічні межі речень [4, 7, 12]. Основним деструктивним фактором такої фрагментації є феномен «невидимого займенника»: у фінансових та юридичних текстах ключові суб'єкти (назви корпорацій) та часові рамки часто відокремлені від конкретних числових показників декількома абзацами. Коли механічний чанкер розділяє ці елементи, вилучений фрагмент містить дані, але втрачає зв'язок із сутністю, до якої вони відносяться, що спотворює векторні представлення (ембединги) та провокує фактологічні галюцинації під час генерації відповіді.

Для подолання цих обмежень запропоновано перехід до архітектури Advanced RAG (рисунок 1), яка трансформує пасивну базу знань у динамічне «середовище виконання знань» (Knowledge Runtime).

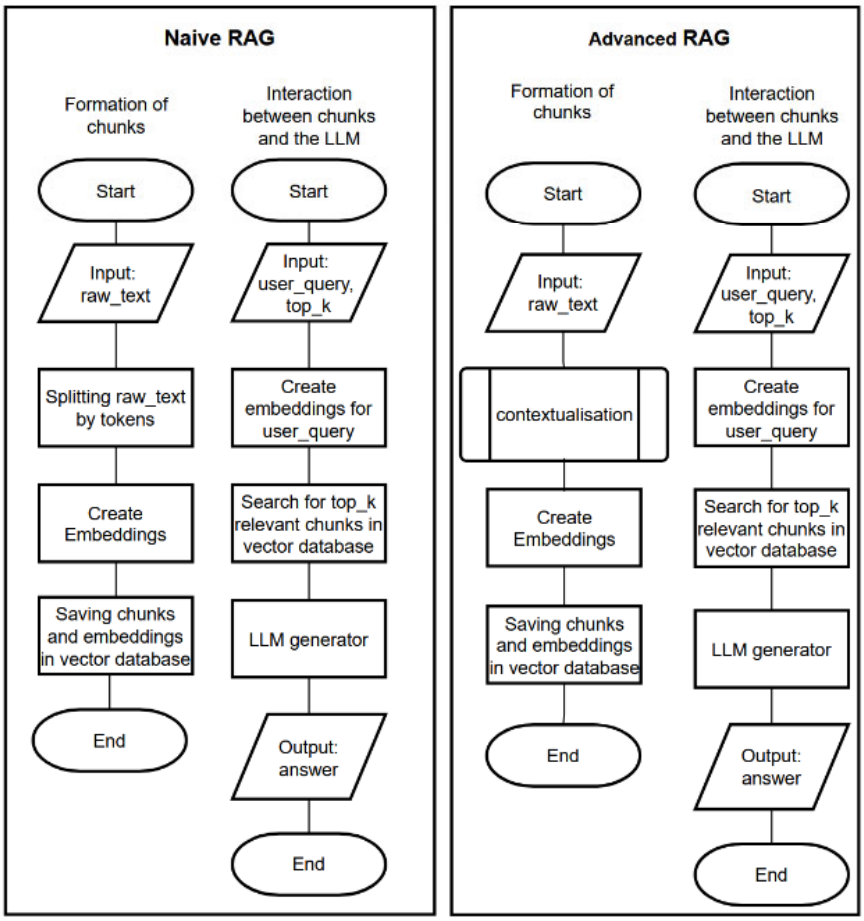

Рисунок 1. Naive і Advanced RAG архітектури

ISSN 2786-6025 Online

В основі методу лежить контекстний пошук (Contextual Retrieval) [9, 10], де кожен фрагмент тексту перед індексацією проходить етап програмного збагачення ситуативним макро-префіксом. Процес обробки розділено на дві фази: Input & Enrichment (Pre-computation), де здійснюється аналіз та збагачення даних, та Retrieval & Synthesis (Runtime), де проводиться семантичний пошук серед збагачених векторів [9, 15]. Це забезпечує збереження глобальної логіки документа навіть у межах дрібних ізольованих сегментів. Технічна реалізація підходу використовує модель gpt-5-nano як контекстуалізатор, що генерує стислий опис ролі кожного фрагмента в структурі документа. Для оптимізації обчислювальних витрат та збереження тяглості оповіді запроваджено механізм «ковзного вікна», що підтримує чергу з трьох останніх описів попередніх сегментів. Процес генерації ситуативного префікса P_i для поточного фрагмента c_i математично описується як:

$$P_i = LLM(S_{i-k}^{i-1}, c_i) \quad (1)$$

де S_{i-k}^{i-1} – масив префіксів попередніх k фрагментів (де k, дорівнює 3).

Програмна логіка модуля передбачає формування префікса за суворим форматом: [Type] Entity Period: Specific Topics, де вказується тип інформації (наприклад, [Table], [CEO Statement], [MD&A]), назва компанії, фіскальний період та 2-4 специфічні категорії даних.

Фінальне векторне представлення збагаченого фрагмента F_{final}(i) обчислюється як результат векторизації (модель text-embedding-3-large) конкатенованого тексту префікса та оригіналу [17]:

$$F_{final}(i) = Embed(P_i \oplus c_i)[17] \quad (2)$$

де Embed – функція векторизації (модель text-embedding-3-large); $\oplus$ – операція текстової конкатенації.

Така архітектура дозволяє системі розв'язувати задачі кореференції ще на етапі створення індексу. Для промислового масштабування рішення реалізовано асинхронну пакетну обробку (batch size = 30) із використанням семафорів для контролю конкурентних запитів [31]. Також інтегровано алгоритм експоненційної затримки (exponential backoff) для стабільної роботи з API у випадках виникнення помилок обмеження частоти запитів [32].

Експериментальна оцінка ефективності розробленого підходу проводилася на наборі даних FinanceBench, що включає 50 складних багатоступеневих фінансових запитів, які потребують глибокого структурного аналізу документів [18, 19, 21]. Оцінювання здійснювалося за допомогою автоматизованого фреймворку RAGAS, що використовує парадигму «LLM-as-a-

ISSN 2786-6025 Online

judge» для вимірювання точності вилучення та генерації [22, 23, 24]. Проведені експериментальні дослідження дозволили отримати результати, що зазначені у таблиці 2. Порівняльний аналіз із базовою архітектурою Naive RAG продемонстрував суттєве покращення ключових метрик. Зокрема, показник Context Recall зріс на 35,46% (з 0.2067 до 0.2800) інші показники подані у таблиці 2, що свідчить про успішне вилучення параметрів із розірваних фінансових таблиць [21, 28].

Покращення метрики Context Precision на 37,99% підтверджує, що додавання ситуативного префікса дозволяє математично групувати вектори у відповідні латентні околиці, мінімізуючи «лексичні колізії» та вилучення нерелевантного контексту [24, 25]. Крім того, спостерігається зростання показника Faithfulness на 7,88%, що прямо корелює зі зниженням ймовірності фабрикації фактів мовною моделлю через відсутність критичних маркерів у вхідних даних [1, 3]. Загальна точність відповідей (Answer Correctness) зросла на 17%, що підтверджує доцільність використання контекстного збагачення для створення надійних інтелектуальних систем корпоративного рівня у високо-точних доменах.

Таблиця 2

Порівняння метрик для базової та вдосконаленої архітектури.

Категорія метрики	**Naive RAG**	**Advanced RAG**	**Відносне покращення**
Context Recall	0.2067	0.2800	35.46%
Context Precision	0.2803	0.3868	37.99%
Faithfulness	0.6137	0.6621	7.88%
Answer Correctness	0.3063	0.3584	17%

Висновки. Результати проведеного дослідження підтверджують, що проблема фактологічних галюцинацій та деградація контексту залишається технічним обмеженням для інтелектуалізації корпоративного сектору, де точність є безальтернативною вимогою. Кількісний аналіз продемонстрував, що стандартні підходи Naive RAG демонструють низьку ефективність через механічну фрагментацію, яка руйнує семантичні зв'язки та зашумлює векторний простір неоднозначними даними. Запропонована архітектура на

ISSN 2786-6025 Online

основі методу контекстного пошуку (Contextual Retrieval) дозволяє нейтралізувати ці недоліки шляхом програмного «приварювання» синтезованого макро-контексту до кожного інформаційного вузла. Це остаточно вирішує проблему «невидимого займенника», забезпечуючи контекстуалізацію, розрізнені факти у конкретне ситуативне середовище документа.

Емпіричне підтвердження успішності запропонованого підходу показало зростання показників Context Recall на 35,46% та Context Precision на 37,99%. Це свідчить про радикальне покращення повноти та точності представлення даних. Така якість вилучення контексту спричиняє вищий рівень безпеки та надійності інтелектуальних систем: підвищення Faithfulness на 7,88% та загальної правильності відповідей на 17% мінімізує ризики фабрикації фактів. Важливо, що для суворого фінансового аудиту та юридичного консалтингу така цілісність структури створює прозорий ланцюжок походження даних до оригінальних джерел. Враховуючи економічну перевагу RAG над використанням моделей із надвеликим контекстним вікном для терабайтних баз знань, впровадження контекстного збагачення стає стратегічним завданням для побудови стійких та масштабованих корпоративних екосистем штучного інтелекту [11, 26].

Література:

1. Хуанг Л. та ін. (2023). Огляд галюцинацій у великих мовних моделях: принципи, таксономія, виклики та відкриті питання. arXiv preprint. URL: https://arxiv.org/html/2311.05232v2

2. Велика мовна модель. (2024). Вікіпедія, Вільна енциклопедія. URL: https://en.wikipedia.org/wiki/Large_language_model

3. Казларіс І., Антоніу Е., Діамантарас К., Братсас К. (2025). Від ілюзії до розуміння: таксономічний огляд методів пом'якшення галюцинацій у LLM. AI, 6(10), 260. DOI: https://doi.org/10.3390/ai6100260

4. Обмеження чанкінгу та пошуку в QA. (2024). Reddit (r/Rag). URL: https://www.reddit.com/r/Rag/comments/1jh2xgs/limitations_of_chunking_and_retrieval_in_qa

5. Льюїс П. та ін. (2020). Пошукова доповнена генерація для складних завдань NLP. Advances in Neural Information Processing Systems, 33, 9459–9474. URL: https://arxiv.org/pdf/2005.11401

6. Косинусна подібність. (2024). Вікіпедія, Вільна енциклопедія. URL: https://en.wikipedia.org/wiki/Cosine_similarity

7. Найкращі стратегії чанкінгу для RAG у 2025 році. (2025). Firecrawl. URL: https://www.firecrawl.dev/blog/best-chunking-strategies-rag-2025

8. Парадигма чанкінгу: рекурсивна семантика для оптимізації RAG. (2025). Матеріали 8-ї Міжнародної конференції з обробки природної мови та мовлення (ICNLSP-2025). URL: https://aclanthology.org/2025.icnlsp-1.15.pdf

9. Контекстний пошук. (2024). Новини Anthropic. URL: https://www.anthropic.com/news/contextual-retrieval

10. Контекстний пошук. (2024). Anthropic Engineering. URL: https://www.anthropic.com/engineering/contextual-retrieval

ISSN 2786-6025 Online

11. Махарджан А., Ядав У. (2026). Чанкінг, пошук та переранжування: емпірична оцінка архітектур RAG для відповідей на запитання за програмними документами. arXiv preprint. URL: https://arxiv.org/html/2601.15457v1

12. Семантичний чанкінг. (2024). Superlinked VectorHub. URL: https://superlinked.com/vectorhub/articles/semantic-chunking

13. Гао Й. та ін. (2023). Пошукова доповнена генерація для великих мовних моделей: огляд. arXiv preprint. URL: https://arxiv.org/pdf/2312.10997

14. Алгоритм семантичного чанкінгу. (2024). EmergentMind. URL: https://www.emergentmind.com/topics/semantic-chunking-algorithm

15. Навчальний посібник з контекстного пошуку Anthropic. (2024). DataCamp. URL: https://www.datacamp.com/tutorial/contextual-retrieval-anthropic

16. Ранжувальники в RAG. (2024). Pinecone. URL: https://www.pinecone.io/learn/series/rag/rerankers/

17. Сансев'єро О. (2024). Ембедінги речень. Блог Hackerllama. URL: https://osanseviero.github.io/hackerllama/blog/posts/sentence_embeddings2/

18. Набір даних PatronusAI/financebench. (2023). Hugging Face. URL: https://huggingface.co/datasets/PatronusAI/financebench

19. Іслам П. та ін. (2023). FinanceBench: новий еталон для відповідей на фінансові запитання. arXiv preprint. URL: https://arxiv.org/pdf/2311.11944v1

20. Документація метрик. (2024). RAGAS. URL: https://docs.ragas.io/en/latest/concepts/metrics/index.html

21. Огляд метрик. (2024). RAGAS. URL: https://docs.ragas.io/en/latest/concepts/metrics/overview.html

22. Точність контексту (Context Precision). (2024). RAGAS. URL: https://docs.ragas.io/en/latest/concepts/metrics/context_precision.html

23. Отримання кращих відповідей RAG за допомогою RAGAS. (2024). Блог Redis. URL: https://redis.io/blog/get-better-rag-responses-with-ragas

24. Одхіамбо Т. (2024). Найефективніший підхід до RAG на сьогодні: контекстний пошук Anthropic та гібридний пошук. Medium. URL: https://medium.com/@tom.odhiambo/the-most-effective-rag-approach-to-date-anthropics-contextual-retrieval-and-hybrid-search

25. Метрики оцінювання RAG. (2024). Блог Confident AI. URL: https://www.confident-ai.com/blog/rag-evaluation-metrics

26. Саченко А. та ін. (2024). Оцінка ансамблевих моделей машинного навчання для систем рекомендацій фільмів. CEUR Workshop Proceedings, 3711, 273–286.

27. Про схвалення Концепції розвитку штучного інтелекту в Україні : Розпорядження Кабінету Міністрів України від 2 грудня 2020 р. № 1556-р. (2020). URL: https://zakon.rada.gov.ua/laws/show/1556-2020-р

28. Документація Asyncio: корутини та завдання. (б.р.). Python Software Foundation. URL: https://docs.python.org/3/library/asyncio-task.html

29. Експоненціальна затримка та джиттер. (б.р.). Блог архітектури AWS. URL: https://aws.amazon.com/blogs/architecture/exponential-backoff-and-jitter/

30. Романенко Є.О. Місце та роль Державної комісії з регулювання ринків фінансових послуг України в регулюванні діяльності недержавних пенсійних фондів // Економіка і регіон. Науковий вісник Полтавського національного технічного університету імені Юрія Кондратюка. №1(2) 2004, с.163-166.

31. Про загальнообов'язкове державне пенсійне страхування. Закон України від 9 липня 2003 року № 1058-IV.

ISSN 2786-6025 Online

32. Проект Закону про загальнообов'язкове накопичувальне пенсійне забезпечення. №2683 від 27.12.2019. http://w1.c1.rada.gov.ua/pls/zweb2/webproc4_1?pf3511=67794

References:

1. Huang, L., et al. (2023). A survey on hallucinations in large language models: principles, taxonomy, challenges, and open questions. arXiv preprint. Retrieved from https://arxiv.org /html/2311.05232v2.

2. Velyka movna model [Large language model]. (2024). In Wikipedia, the Free Encyclopedia. Retrieved from https://en.wikipedia.org/wiki/Large_language_model.

3. Kazlaris, I., Antoniou, E., Diamantaras, K., & Bratsas, K. (2025). Vid iliuzii do rozuminnia: taksonomichnyi ohliad metodiv pom'iakshennia haliutsynatsii u LLM [From illusion to understanding: a taxonomic survey of hallucination mitigation methods in LLMs]. *AI*, 6(10), 260. DOI: https://doi.org/10.3390/ai6100260.

4. Limitations of chunking and retrieval in QA. (2024). Online forum post. Retrieved from https://www.reddit.com/r/Rag/comments/1jh2xgs/limitations_of_chunking_and_retrieval_in_qa

5. Lewis, P., et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33, 9459–9474. https://arxiv.org/ pdf/2005.11401.

6. Kosynusna podibnist [Cosine similarity]. (2024). In Wikipedia, the Free Encyclopedia. Retrieved from https://en.wikipedia.org/wiki/Cosine_similarity [in Ukrainian].

7. Naikrashchi stratehii chankinhu dlia RAG u 2025 rotsi [Best chunking strategies for RAG in 2025]. (2025). Retrieved from https://www.firecrawl.dev/blog/best-chunking-strategies-rag-2025.

8. Paradyhma chankinhu: rekursyvna semantyka dlia optymizatsii RAG [The chunking paradigm: recursive semantics for RAG optimization]. (2025). In Proceedings of the 8th International Conference on Natural Language and Speech Processing (pp. 109–111). Retrieved from https://aclanthology.org/2025.icnlsp-1.15.pdf.

9. Kontekstnyi poshuk [Contextual retrieval]. (2024). *Novyny Anthropic [Anthropic News]*. Retrieved from https://www.anthropic.com/news/contextual-retrieval.

10. Kontekstnyi poshuk [Contextual retrieval]. (2024). *Anthropic Engineering*. Retrieved from https://www.anthropic.com/engineering/contextual-retrieval.

11. Maharjan, A., & Yadav, U. (2026). Chankinh, poshuk ta pereranzhuvannia: empirychna otsinka arkhitektur RAG dlia vidpovidei na zapytannia za prohramnymy dokumentamy [Chunking, retrieval, and reranking: an empirical evaluation of RAG architectures for software documentation QA]. *arXiv preprint*. Retrieved from https://arxiv.org/html/2601.15457v1.

12. Semantic chunking [Semantic chunking]. (2024). Retrieved from https://superlinked. com/vectorhub/articles/semantic-chunking.

13. Gao, Y., et al. (2023). Retrieval-augmented generation for large language models: A survey. arXiv. https://arxiv.org/pdf/2312.10997.

14. Semantic chunking algorithm [Algorithm of semantic chunking]. (2024). Retrieved from https://www.emergentmind.com/topics/semantic-chunking-algorithm.

15. Anthropic's contextual retrieval tutorial [Anthropic's contextual retrieval tutorial]. (2024). Retrieved from https://www.datacamp.com/tutorial/contextual-retrieval-anthropic.

16. Ranzhuvalnyky v RAG [Rerankers in RAG]. (2024). *Pinecone*. Retrieved from https://www.pinecone.io/learn/series/rag/rerankers/.

17. Sanseviero, O. (2024). Sentence embeddings. Hackerllama Blog. https://osanseviero. github.io/hackerllama/blog/posts/sentence_embeddings2/.

ISSN 2786-6025 Online

18. PatronusAI/financebench dataset. (2023). Hugging Face. Retrieved from https://huggingface.co/datasets/PatronusAI/financebench

19. Islam, P., et al. (2023). FinanceBench: A new benchmark for financial question answering. arXiv. https://arxiv.org/pdf/2311.11944v1.

20. Dokumentatsiia metryk [Metrics documentation]. (2024). *RAGAS*. Retrieved from https://docs.ragas.io/en/latest/concepts/metrics/index.html.

21. Ohliad metryk [Metrics overview]. (2024). *RAGAS*. Retrieved from https://docs.ragas.io/en/latest/concepts/metrics/overview.html.

22. Tochnist kontekstu (Context Precision) [Context Precision]. (2024). *RAGAS*. Retrieved from https://docs.ragas.io/en/latest/concepts/metrics/context_precision.html.

23. Otrymannia krashchykh vidpovidei RAG za dopomohoiu RAGAS [Getting better RAG responses with RAGAS]. (2024). *Bloh Redis [Redis Blog]*. Retrieved from https://redis.io/blog/get-better-rag-responses-with-ragas.

24. Odhiambo, T. (2024). Naiefektyvnishyi pidkhid do RAG na sohodni: kontekstnyiposhuk Anthropic ta hibrydnyi poshuk [The most effective RAG approach to date: Anthropic's cont-extual retrieval and hybrid search]. Blog post. Retrieved from https://medium.com/ @tom.odhiambo/the-most-effective-rag-approach-to-date-anthropics-contextual-retrieval-and-hybrid-search.

25. RAG evaluation metrics. (2024). Confident AI Blog. Retrieved from https://www .confident-ai.com/blog/rag-evaluation-metrics

26. Sachenko, A., Halias, Yu., Lipianina-Honcharenko, K., Hladii, H., Koval, V., & Lendiuk, T. (2024). Otsinka ansamblevykh modelei mashynnoho navchannia dlia system rekomendatsii filmiv [Evaluation of ensemble machine learning models for movie recommendation systems]. *CEUR Workshop Proceedings, 3711*, 273–286.

27. Cabinet of Ministers of Ukraine. (2020). *Pro skhvalennia Kontseptsii rozvytku shtuchnoho intelektu v Ukraini [On approval of the Concept of artificial intelligence development in Ukraine]* (Order No. 1556-r). Retrieved from https://zakon.rada.gov.ua/laws/show/1556-2020-p [in Ukrainian].

28. Dokumentatsiia Asyncio: korutyny ta zavdannia [Asyncio documentation: coroutines and tasks]. (n.d.). Python Software Foundation. Retrieved from https://docs.python.org/3/library/asyncio-task.html.

29. Eksponentsiialna zatrymka ta dzhytter [Exponential backoff and jitter]. (n.d.). AWS Architecture Blog. Retrieved from https://aws.amazon.com/blogs/architecture/exponential-backoff-and-jitter/.

30. Romanenko, Ye.O. (2004). Mistse ta rol Derzhavnoi komisii z rehuliuvannia rynkiv finansovykh posluh Ukrainy v rehuliuvanni diialnosti nederzhavnykh pensiinykh fondiv [The place and role of the State Commission for Regulation of Financial Services Markets of Ukraine in regulating the activities of non-state pension funds]. *Ekonomika i rehion*, 1(2), 163-166.

31. Pro zahalnooboviazkove derzhavne pensiine strakhuvannia [On compulsory state pension insurance]. (2003). Zakon Ukrainy No. 1058-IV. Retrieved from https://zakon.rada.gov.ua/laws/show/1058-15 [in Ukrainian].

32. Proekt Zakonu pro zahalnooboviazkove nakopychuvalne pensiine zabezpechennia [Draft Law on compulsory accumulative pension provision]. №2683 vid 27.12.2019. (2019). *Zakon.rada.gov.ua*. Retrieved from http://w1.c1.rada.gov.ua/pls/zweb2/webproc4_1?pf3511=67794 [in Ukrainian].

Дата першого надходження статті до видання: 11.04.2026
Дата прийняття статті до друку після рецензування: 26.04.2026

Журнал

«Наука і техніка сьогодні»

Випуск № 4(58) 2026

Формат 60х90/8. Папір офсетний.
Гарнітура Times New Roman.
Ум. друк. арк. 8,2. Наклад 100 прим.

Видавець:
Громадська наукова організація «Всеукраїнська асамблея докторів наук з державного управління»
Свідоцтво серія ДК №4957 від 18.08.2015 р., Андріївський узвіз, буд.11, оф 68, м. Київ, 04070.