УДК 004.85:004.77

С.П. Пазьміно-Боскес
А.В. Лосенко

ЗАСТОСУВАННЯ ВЕБ-ПОШУКУ ТА ПРОТОКОЛУ МОДЕЛЬНОГО КОНТЕКСТУ В АРХІТЕКТУРАХ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ

Вінницький національний технічний університет

Анотація

Проаналізовано методи динамічної наповнення контексту великих мовних моделей через інтеграцію інструментів веб-пошуку. Розглянуто архітектурну специфіку протоколу Model Context Protocol (MCP) та сервісів пошукового обґрунтування Gemini Grounding для підвищення фактичної точності генерації. Визначено підходи до оптимізації використання токенів та верифікації джерел. Запропоновано розширену класифікацію стратегій формування пошукових запитів та методів ранжування джерел.

Ключові слова: великі мовні моделі, RAG, веб-пошук, Model Context Protocol, Gemini Grounding, Query Decomposition, верифікація джерел.

Abstract

The methods of dynamic contextualization of large language models through the integration of web search tools are analyzed. The architectural specificity of the Model Context Protocol (MCP) and Gemini Grounding services for increasing the factual accuracy of generation is considered. Approaches to token usage optimization and source verification are defined. An extended classification of query formation strategies and source ranking methods is proposed.

Keywords: large language models, RAG, web search, Model Context Protocol, Gemini Grounding, Query Decomposition, source verification.

Вступ

Обмеженість параметричної пам'яті великих мовних моделей (Large Language Model, LLM) датою завершення навчання обумовлює необхідність впровадження механізмів генерації, доповненої пошуком (Retrieval-Augmented Generation, RAG). Використання відкритої мережі Інтернет як зовнішнього джерела даних дозволяє мінімізувати галюцинації та забезпечити актуальність відповідей у реальному часі [1]. Актуальним завданням є стандартизація взаємодії між LLM та пошуковими сервісами, що реалізується через новітній протокол модельного контексту (Model Context Protocol, MCP) та систем пошукового обґрунтування.

Сучасні дослідження демонструють, що частота галюцинацій у LLM без зовнішнього заземлення сягає 27–46% для фактологічних запитів у предметних областях з динамічно змінюваними даними [2]. Механізми RAG знижують цей показник до 8–12%, тоді як гібридні підходи, що поєднують параметричну пам'ять моделі з живим веб-пошуком, демонструють похибку менше 5%. Ключовою проблемою залишається не лише отримання релевантних даних, а й їх семантична інтеграція в контекст генерації без втрати когерентності відповіді.

Архітектурна реалізація через Model Context Protocol (MCP)

Протокол MCP, представлений у листопаді 2024 року, запроваджує уніфікований стандарт підключення моделей до зовнішніх джерел даних за принципом «клієнт-сервер» [3]. Ключовою перевагою MCP є уніфікація доступу до пошукових систем: розробнику не потрібно створювати окремі конектори для кожного провайдера (Google, Brave, Bing, Exa), оскільки протокол забезпечує єдиний інтерфейс взаємодії незалежно від конкретного сервісу.

З точки зору ефективності використання контекстного вікна, оновлення MCP Tool Search (січень 2026 р.) реалізує принцип відкладеного завантаження (lazy loading) специфікацій інструментів — їх опис підвантажується лише в момент фактичного виклику, а не передається

моделі на початку кожної сесії. За даними Anthropic, це дозволило скоротити споживання токенів на 85% у сценаріях із великою кількістю підключених MCP-серверів [3]. Подібний підхід узгоджується із загальною тенденцією до динамічного управління контекстом у агентних системах, описаною в роботах з оптимізації RAG-систем [4].

Окремим напрямом є нормалізація веб-контенту перед його подачею до моделі. MCP-сервери типу Fetch або Firecrawl виконують перетворення HTML-розмітки у структурований Markdown-формат на стороні сервера ще до потрапляння тексту до контекстного вікна. Це дозволяє усунути розмітковий шум (теги навігації, рекламні блоки, скрипти) та зосередити токени виключно на семантично значущому вмісті сторінки. Дослідження у сфері препроцесингу документів для RAG підтверджують, що якість нормалізації вхідного тексту безпосередньо корелює з точністю відповіді моделі [5].

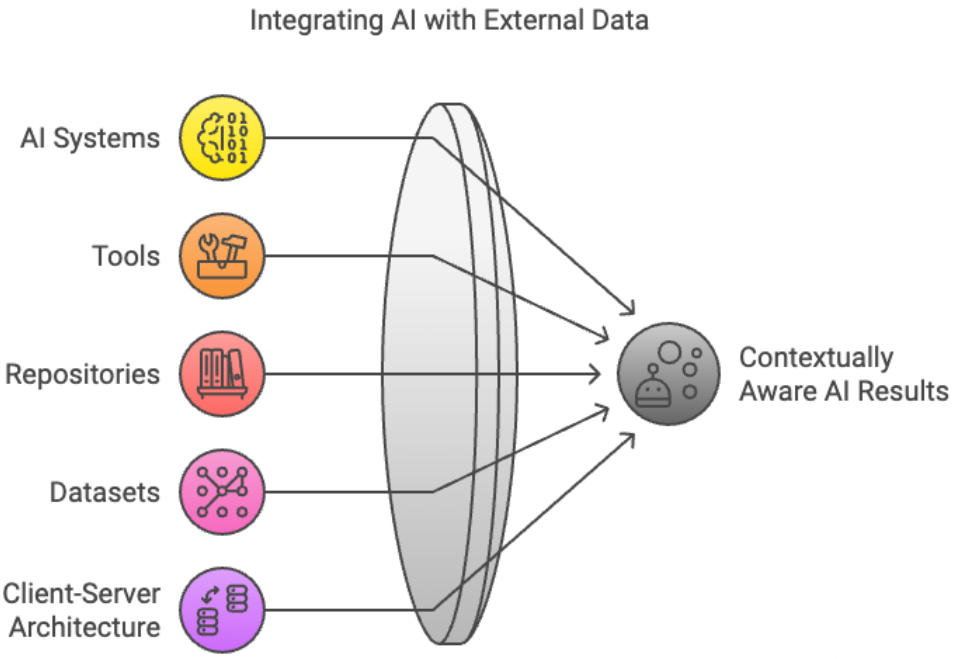

Рис.1 Інструменти для покращення контексту великої мовної моделі

Технологія Gemini Grounding

Технологія пошукового обґрунтування (Grounding) від Google інтегрує пошукову систему безпосередньо в ітераційний цикл генерації моделі [6]. На відміну від класичного RAG, де розробник самостійно будує пошуковий механізм, Gemini Grounding самостійно визначає потребу в зовнішніх даних (grounding data) та синтезує відповідь з цитуванням джерел.

Використання динамічних порогів (dynamic grounding thresholds) дозволяє активувати пошук лише за умови низької впевненості моделі у власних знаннях, що суттєво знижує латентність системи. Результати генерації супроводжуються об'єктом groundingMetadata, що забезпечує повну простежуваність та верифікацію тверджень.

З практичної точки зору, Gemini Grounding реалізує тісніший зв'язок між пошуком і генерацією порівняно з класичними RAG-системами: замість одноразового отримання документів до початку генерації, модель може ініціювати додаткові пошукові запити безпосередньо в процесі формування відповіді, якщо проміжний результат виявляється недостатньо обґрунтованим. Такий підхід концептуально близький до методу IRCoT (Interleaving Retrieval with Chain-of-Thought), де пошук і міркування чергуються ітераційно для вирішення складних багатокрокових запитів [7].

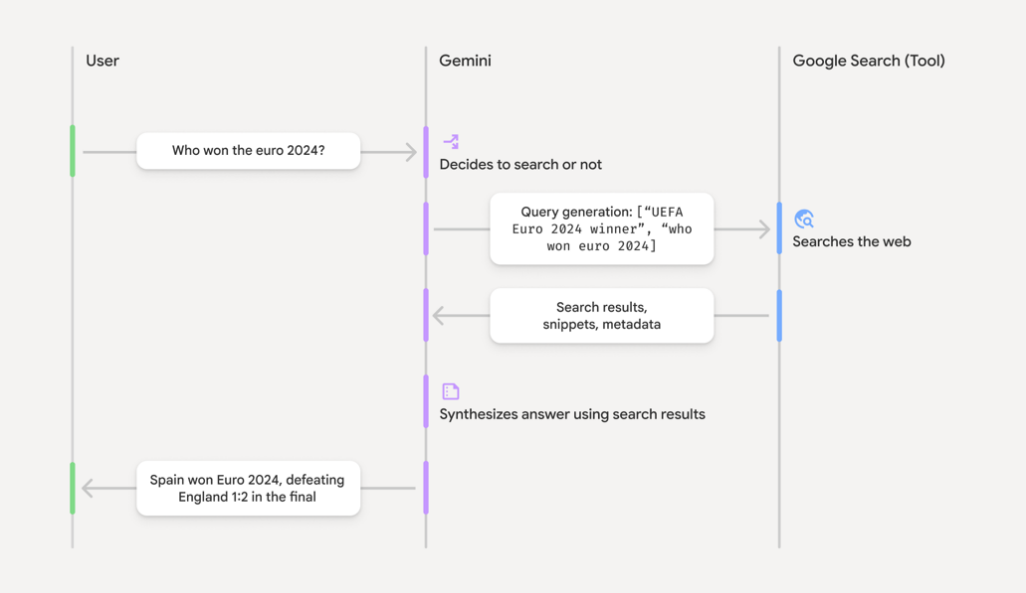

Рис.2 Принцип роботи Google Grounding Search у поєднанні з Gemini

Стратегії оптимізації пошукових запитів

Ефективність збагачення відповіді LLM через веб-пошук визначається не лише архітектурою пошукового інструменту, а й якістю формування пошукових запитів. Окреслимо деякі з відомих стратегій [7, 8].

Query Decomposition — декомпозиція складного користувацького запиту на ряд атомарних підзапитів. Модель генерує від 2 до 5 цільових запитів, кожен з яких покриває окремий аспект вихідного питання. Така стратегія показує перевагу при обробці багатокрокових аналітичних запитів.

Iterative Refinement — ітераційне уточнення запиту на основі проміжних результатів пошуку. Після аналізу першого набору результатів модель формує уточнюючі запити, що дозволяє подолати неоднозначність термінів та підвищити точність джерел.

HyDE (Hypothetical Document Embeddings) — генерація гіпотетичного документа-відповіді до фактичного пошуку, який потім використовується як семантичний вектор для знаходження релевантних джерел.

Верифікація та ранжування джерел

Критичним компонентом архітектури є шар постобробки пошукових результатів. Він включає декілька взаємодоповнюючих механізмів:

- Перехресна верифікація — зіставлення тверджень між декількома незалежними джерелами для виявлення суперечностей.
- Часова фільтрація — автоматична пріоритизація джерел на основі дати публікації, що є особливо важливим для запитів про поточні події.
- Авторитетність домену — присвоєння ваги джерелу на основі TLD, індексу цитування або приналежності до верифікованих реєстрів (.gov, .edu, peer-reviewed).

Сучасні реалізації, зокрема на базі MCP-серверів типу Brave Search або Tavily, дозволяють отримувати разом із текстом метадані достовірності (credibility score), що автоматизує процес ранжування безпосередньо на рівні протоколу.

Висновки

Впровадження MCP та систем пошукового обґрунтування типу Gemini Grounding знаменує перехід від жорстко заданих RAG-систем до гнучких агентних архітектур. Це дозволяє досягти

стабільної точності генерації в умовах динамічної зміни інформаційного середовища при одночасному зниженні інфраструктурних витрат на обробку контексту.

Перспективним напрямом є розвиток мультиагентних архітектур, де спеціалізований агент-пошуковик відокремлений від агента-синтезатора відповіді. Такий поділ відповідальності дозволяє паралелізувати виконання пошукових підзапитів та незалежно масштабувати кожен компонент системи. Інтеграція MCP як транспортного шару в подібних архітектурах забезпечує стандартизований обмін структурованими результатами між агентами без прив'язки до конкретного провайдера пошуку, що суттєво підвищує відтворюваність та портативність рішень.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Lewis P. et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. Advances in Neural Information Processing Systems. – 2020. – Vol. 33. – P. 9459–9474.
2. Huang L. et al. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. ACM Transactions on Information Systems. – 2025. – Vol. 43, No. 2. – Article 42. DOI: 10.1145/3703155.
3. Anthropic. Model Context Protocol Specification [Електронний ресурс]. – Режим доступу: https://modelcontextprotocol.io/
4. Gao Y. et al. Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv:2312.10997. – 2024 (v5). – [Електронний ресурс]. – Режим доступу: https://arxiv.org/abs/2312.10997
5. Li Y. et al. Mitigating Hallucination in Large Language Models: An Application-Oriented Survey on RAG, Reasoning, and Agentic Systems. – arXiv:2510.24476, 2025. – [Електронний ресурс]. – Режим доступу: https://arxiv.org/abs/2510.24476
6. Google Cloud. Grounding with Google Search in Gemini [Електронний ресурс]. – Режим доступу: https://ai.google.dev/gemini-api/docs/google-search
7. Trivedi H. et al. Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions. Proceedings of the 61st Annual Meeting of the ACL. – Toronto, 2023. – P. 10014–10037.
8. Gao L. et al. Precise Zero-Shot Dense Retrieval without Relevance Labels. Proceedings of the 61st Annual Meeting of the ACL. – Toronto, 2023. – P. 1762–1777.

Пазьміно Сауль Патрісіо *– студент групи 2ІСТ-24б, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м. Вінниця.*

Лосенко Арсен Володимирович – PhD, асистент кафедри системного аналізу та інформаційних технологій, Вінницький національний технічний університет, м. Вінниця.

B. Pazmino Saul P. – *student of Faculty of Intellectual Information Technologies and Automation, 2IST-24b, Vinnytsia National Technical University, Vinnytsia.*

Losenko Arsen V. – PhD, assistant of the Department of System Analysis and Information Technologies, Vinnytsia National Technical University, Vinnytsia.