УДК 004.89:004.934:005.53 **DOI:** 10.31673/2412-9070.2026.028906

О. Я. КІС, асистент;
ORCID: 0009-0006-9174-2588
М. С. КЛИМЕНКО, в. о. завідувача науково-дослідної частини,
ORCID: 0000-0003-4433-6641
Державний університет інформаційно-комунікаційних технологій, Київ

МЕТОДИКА ЗАСТОСУВАННЯ LLM ДЛЯ ПІДТРИМКИ ПРИЙНЯТТЯ УПРАВЛІНСЬКИХ РІШЕНЬ

У статті представлена методика оптимізації великих мовних моделей (LLM) для обробки надвеликих масивів контексту для підтримки прийняття управлінських рішень. Досліджується проблема «lost-in-the-middle» та спричинена нею деградація точності вилучення фактів при збільшенні обсягу вхідних даних у запиті до LLM понад 100 000 токенів. Запропоновано гібридну концепцію, що базується на поєднанні технічного аудиту моделі та методів інженерії запитів (Prompt Engineering), зокрема динамічної ін'єкції аналітичних моделей (DAP) та передавання вже обчислених статистичних параметрів (регресійних коефіцієнтів) у структурованому текстовому форматі (JSON, Markdown). Визначено систему критичних технічних метрик (TTFT, TPOT, Needle In A Haystack), що безпосередньо корелюють з оперативністю та валідністю управлінських рішень. Запропонована методологія структурованого передавання даних дає змогу мінімізувати арифметичні галюцинації та зменшити когнітивне навантаження моделі. Використання засобів інженерії запитів дозволяє трансформувати LLM з інструменту генерації тексту на повноцінний інтерпретатор складних аналітичних моделей без необхідності донавчання.

Ключові слова: великі мовні моделі (LLM), інженерія запитів (Prompt Engineering), RAG-emulation, система підтримки прийняття рішень, управлінські рішення, аналітична модель, інженерія запитів, RAG, JSON, Markdown.

Вступ

Сучасна парадигма цифрової трансформації управління вимагає переходу до високотехнологічних методів обробки надвеликих обсягів неструктурованої інформації. Стрімкий розвиток великих мовних моделей (LLM - Large Language Models), які за визначенням є нейромережевими архітектурами, навченими на гігантських текстових масивах для прогнозування наступного токена [1], відкрив безпрецедентні можливості для автоматизації аналітичних процесів. Зокрема, LLM стають ядром сучасних систем підтримки прийняття рішень (DSS - Decision Support Systems), що допомагають аналізувати альтернативи та прогнозувати наслідки управлінських кроків [2].

Відбувається стрімке удосконалення основних характеристик, що впливають на ефективний обсяг інформації для обробки у промпті: розширення контекстних вікон до мільйонів токенів та збільшення загального розміру моделі, відповідні модифікації механізму уваги та позиційне кодування для роботи із великим контекстом. Однак, незважаючи на це, архітектурні обмеження трансформерів щодо точності математичних обчислень залишаються критичним бар'єром. Ключовою проблемою є нездатність моделей до проведення складних статистичних розрахунків за даними запиту без відповідного донавчання на релевантних прикладах обробки інформації. Моделі часто демонструють явища арифметичних галюцинацій при спробі одночасно вилучати дані та проводити регресійний аналіз. Ця робота присвячена розробці методики, яка покликана підвищити аналітичну точність на основі засобів інженерії запитів (Prompt Engineering) LLM, виступаючи інтерпретатором попередньо розрахованих параметрів у форматах JSON та Markdown.

Аналіз літературних даних та постановка проблеми

У роботі Парк та ін. [3] розглядається проблема емуляції генерації з доповненим пошуком (RAG - Retrieval-Augmented Generation) - технології, що дозволяє моделі звертатися до зовнішніх значень або специфічних сегментів контексту для підвищення точності відповідей [4]. Автори доводять, що просте збільшення обсягу даних не гарантує успішного міркування (multi-hop reasoning), якщо релевантні факти розсіяні.

Проблематику системної оптимізації запитів розглянуто у працях Сторчак К. П. та ін. Автори доводять, що ефективність інтелектуальних систем безпосередньо залежить від якості інструктивного дизайну, який охоплює не лише текстове формулювання, а й логічну організацію запиту. Це підтверджує роль Prompt Engineering як одного з ключових засобів керування увагою моделі [5]. Встановлено, що якість результатів прямо залежить від інструктивного дизайну (Instructional Design). Проблема «lost-in-the-middle» підтверджена бенчмарками NovelQA та BABILong [6], вказує на втрату уваги моделі до центру контексту. Це зумовлює потребу в Dynamic Analysis Prompting (DAP) та Chain-of-Thought (CoT) [7].

Використання LLM у якості безпосереднього предиктора або інтелектуального оркестратора обчислювальних процесів є сучасним напрямом досліджень. Наразі можна виокремити низку ключових типів моделювання відповідно до специфіки обробки даних та застосування математичних закономірностей. У *агентному моделюванні* (Agent-Based Modeling, ABM) LLM використовуються для симуляції складної поведінки окремих суб'єктів у соціальних, економічних або екологічних системах, де акцент робиться на реалістичності сценаріїв людської взаємодії [8]. LLM здатні до імітації антропоморфного міркування, адаптації та прийняття рішень, де кожен агент оперує мовною та узагальненою числовою інформацією про стан віртуального середовища. *При побудові універсальних предикторів* (Zero-Shot Time Series Forecasting), які можуть екстраполювати дані у фізиці, фінансах або логістиці, дослідники покладаються на можливості LLM з виявлення статистичних закономірностей без донавчання, що дозволяє токенізувати числові набори даних аналогічно до іншої інформації запиту [9]. *Символьне та чисельне розв'язання математичних задач* використовує LLM для інтелектуальної формалізації описів предметної площини (у математичні рівняння та/або програмний код) для подальшого запуску чисельного інтегрування або оптимізації параметрів. Цей підхід є критичним для інженерних розрахунків та створення «цифрових двійників» оскільки оминає проблему арифметичних галюцинацій: LLM відповідає за логічну структуру моделі, а чисельні розрахунки проводяться спеціалізованими середовищами [10]. Головним недоліком даного підходу є ймовірність генерації невалідного підходу до формалізації, що надаватиме хибний результат й водночас приховає помилку розрахунків.

Метою даної статті є обґрунтування методики на основі інженерії запитів для підвищення достовірності DSS через передавання структурованих наборів даних, що описують ключові показники цільових процесів.

Основна частина

Запропонована методика базується на основі інженерії запитів та передаванні структурованих параметрів, що дозволяє послідовно мінімізувати когнітивні викривлення моделі при роботі з числовими даними (Рис.).

Наукова новизна авторської методики DAP полягає у інтеграції методів багатокрокового міркування із зовнішніми валідованими статистичними моделями для потреб стратегічного менеджменту, що робиться вперше. На відміну від механізму виклику функцій, що є сучасним підходом LLM до застосування відповідних обробників для специфічних форматів інформації, методика орієнтована на великі обсяги структурованих аналітичних даних. Таким чином, досягається трансформація фундаментальної парадигми використання LLM у системах підтримки прийняття рішень: від ролі безпосереднього обчислювача до ролі експертного інтерпретатора структурованих аналітичних даних.

Методика складається із 3 основних фаз, що зумовлено логікою перетворення сирих даних у валідоване управлінське рішення. Кожна фаза виконує специфічну роль у забезпеченні робастності системи, сприяючи по-доланню фундаментальних архітектурних обмежень сучасних трансформерних моделей.

1. Технічний аудит та ідентифікація обмежень (Audit Phase)

Для стабільної роботи аналітичної системи необхідно визначити метрики, що можуть надати кількісну оцінку якості інженерії запитів:

– TTFT (Time To First Token): час очікування початку генерації першого символу. У контексті DSS ця метрика визначає «відчуття» інтерактивності системи. Високий TTFT на великих промптах (понад 128к токенів) може свідчити про перевантаження механізмів уваги, що призводить до затримок у прийнятті критичних рішень у реальному часі [5].

– TPOT (Time Per Output Token): середній час генерації кож-ного наступного символу.

– Стабільність TPOT є індикатором відсутності обчислювального "зациклення" моделі при складних логічних висновках.

– Фактичний обсяг контекстного вікна (Context Window Limit): зростання обсягу токенів у запиті призводить до миттєвої втрати ранньої інформації (FIFO-деградація). Важливо враховувати, що робоче вікно моделі (ефективний контекст) часто менше за заявлене розробником через зростання шуму, оскільки токсичне когнітивне навантаження запиту: надмірна кількість інструкцій у промпті разом із сирими даними вичерпує обчислювальний бюджет уваги моделі, що спричиняє галюцинації [3, 7].

Рис. 1 Алгоритм інтелектуального аналізу за методикою емуляції RAG та DAP

2. Структурована ін'єкція кількісних параметрів (Injection Phase)

Інженерія запитів у цій фазі спрямована на подачу моделі чітко описаних аналітичних даних, що виключає потребу у самостійних обчисленнях великого обсягу під час обробки запиту. У цій роботі пропонується використання регресійних параметрів у якості аналітичного базису: модель отримує готові коефіцієнти нахилу та зміщення, що описують характер тренду. Значення коефіцієнта детермінації слугує для моделі індикатором надійності прогнозу, що дозволяє алгоритмічно відсікати статистично незначущі кореляції під час формування тексту висновку.

Регресійні параметри є результатом роботи зовнішнього валідованого програмного забезпечення (наприклад, Python/Pandas або R). Такий формат забезпечує точність (числа передаються безпосередньо, усуваючи помилки парсингу, нерівномірної уваги) та якісну інтерпретацію, оскільки коефіцієнти вже мають людино-орієнтований опис, що спрямовує модель на коректне застосування значень.

Розглянемо низку поширених форматів представлення структурованих даних у контексті інженерії запитів та передаванні даних до мовних моделей. Кожен з наведених нижче форматів має унікальний набір характеристик, що може робити їх ефективними для різних етапів аналітичного процесу [10-12].

JSON (JavaScript Object Notation): використання цього формату забезпечує сувору типізацію та ієрархічність даних. Моделі легше парсити пари «ключ-значення», оскільки це активує

специфічні патерни обробки, характерні для програмування (Function Calling), що мінімізує змішування чисел із загальним текстом [5].

YAML (YAML Ain't Markup Language): людино орієнтований формат, значно зменшує кількість технічного «шуму» у промпті через використання значущих відступів замість фігурних дужок та лапок. Таким чином, структура даних стає просторово очевидною. Підтримка коментарів дозволяє розробнику надавати моделі додаткові контекстуальні пояснення безпосередньо всередині блоку даних, не порушуючи при цьому синтаксис. Анкори та посилання надають можливість уникати дублювання інформації в межах одного запиту. Основний недолік YAML полягає в неоднозначності стандартів (так звана проблема «NOrway problem»), що створює ризики хибної інтерпретації даних.

XML (eXtensible Markup Language): один із найстаріших форматів структурованих даних, що використовується сучасними інформаційними системами. Основною перевагою XML є семантична виразність: індивідуальні теги дозволяють розмежовувати різні типи даних, інструкції та метадані у межах одного запиту. Документарна орієнтованість робить XML зручною для передавання складних багатокомпонентних об'єктів із розгалуженими атрибутами. XML є стандартом для багатьох корпоративних та державних інформаційних систем, що полегшує інтеграцію LLM-методики в існуючу інфраструктуру без необхідності складної конвертації даних. Проте синтаксична надлишковість XML призводить до зростання накладних витрат з обробки таких запитів. Таким чином, XML краще підходить для фази ін’єкції даних, де важлива максимальна семантична чіткість, але він часто поступається JSON у фазі генерації кінцевих висновків через громіздкість та вимоги до синтаксису.

Markdown (Lighweight Markup Language): застосовується для представлення даних у вигляді структурованих таблиць. Це візуально відокремлює аналітичний блок від інструктивного та дозволяє моделі використовувати механізми просторової уваги для швидкого зіставлення параметрів між стовпцями.

CSV (Comma-Separated Values): дозволяє представляти табличні дані із різнотипним вмістом, є нативним форматом для представлення результатів регресійного аналізу та часових рядів із мінімальним синтаксичним навантаженням. Водночас, наявність гнучких та нестандартизованих варіантів екранування та роздільних символів негативно впливатимуть на однозначність інтерпретації даних.

3. Емуляція RAG та багатокрокове міркування (Reasoning Phase)

Активація логіки Chain-of-Thought (CoT) у цій фазі виступає критичним запобіжником проти когнітивних помилок моделі. Модель не просто генерує висновок, а виконує послідовне зіставлення вилучених текстових фактів із переданими JSON-параметрами.

Процес міркування включає:

– синтез доказів - виділення релевантних фрагментів тексту (тегування), що прямо корелюють із кількісними показниками регресії;

– валідацію контексту - порівняння динамічних трендів із якісними твердженнями в документах для виявлення аномалій.

– логічний висновок - формування фінального управлінського звіту, де кожне твердження підкріплене як параметрично на основі блоку структурованих аналітичних даних, так і фрагментами інформації з неструктурованого контексту. Це дозволяє уникнути арифметичних галюцинацій, оскільки модель оперує вже готовими результатами обчислень [3], [7].

Висновки

Використання методів інженерії запитів дозволяє радикально подолати архітектурні та обчислювальні обмеження сучасних LLM у роботі з надвеликим контекстом. Поєднання алгоритмічно точних розрахунків, інтегрованих через структуровані формати (JSON, YAML, XML, Markdown, CSV) та гнучкої лінгвістичної інтерпретації забезпечує надійність систем підтримки прийняття рішень (DSS). Такий гібридний підхід не лише мінімізує ризики галюцинацій, а й підвищує прозорість логічних ланцюжків, що є фундаментальною вимогою для корпоративного та державного управління.

Запропонована методика є кроком у розвитку LLM-систем підтримки рішень. Вона демонструє, як цілеспрямована інженерія запитів, що поєднується з ін'єкцією структурованих аналітичних моделей, може забезпечити зменшення рівня галюцинацій і помилкових висновків, підвищення відтворюваності результатів комплексних моделювань, водночас зберігаючи адаптивність до застосування альтернативних сценаріїв аналізу.

Подальші дослідження мають бути спрямовані на масштабну емпіричну перевірку та інтеграцію з корпоративними BI-системами.

Внесок авторів

Олександр КІС – опис методики, аналіз джерел, оформлення ілюстрацій, формування висновків; Микита КЛИМЕНКО – опис проблематики дослідження, узагальнення засобів структурованого представлення даних, підготовка літератури.

Декларація про штучний інтелект

Автори не використовували засоби штучного інтелекту для створення матеріалів статті.

Конфлікт інтересів

Автор заявляє про відсутність конфлікту інтересів та підтверджує, що під час підготовки цієї роботи не існувало жодних комерційних, фінансових чи інших взаємовідносин, які могли б бути розцінені як такі, що здатні вплинути на результати дослідження або їх інтерпретацію. Робота виконана відповідно до принципів академічної доброчесності, етичних норм проведення наукових досліджень та вимог редакційної політики щодо запобігання конфлікту інтересів.

Список використаної літератури

1. Brown T. B., Mann B., Ryder N. et al. Language Models are Few-Shot Learners. arXiv, 2020. DOI:10.48550/ARXIV.2005.14165

2. Power D. J. Decision support systems: concepts and resources for managers. 1. publ. Westport, Conn.: Quorum Books, 2002. 251 с. ISBN 978-1-56720-497-1

3. Park J., Atarashi K., Takeuchi K. et al. Emulating Retrieval Augmented Generation via Prompt Engineering for Enhanced Long Context Comprehension in LLMs. arXiv, 2025. DOI:10.48550/arXiv.2502.12462

4. Lewis P., Perez E., Piktus A. et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.. arXiv, 2020. DOI:10.48550/ARXIV.2005.11401

5. Storchak K. P., Mykolaienko V. O., Dovzhenko T. P. Prompt optimization for large language models. Connectivity. Vol. 177, Issue 5. P. 25–31. DOI:10.31673/2412-9070.2025.050843

6. Wang C., Ning R., Pan B. et al. NovelQA: Benchmarking Question Answering on Documents Exceeding 200K Tokens. arXiv, 2024. DOI:10.48550/ARXIV.2403.12766

7. Wei J., Wang X., Schuurmans D. et al. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv, 2022. DOI:10.48550/ARXIV.2201.11903

8. Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative Agents: Interactive Simulacra of Human Behavior. Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23). Association for Computing Machinery, 2, 1–22. DOI: 10.1145/3586183.3606763

9. Gruver, N., Finzi, M., Qiu, S., & Wilson, A. G. (2023). Large language models are zero-shot time series forecasters. Advances in neural information processing systems, 36, 19622-19635. DOI: 10.48550/arXiv.2310.07820

10. Frieder, S., Pinchetti, L., Chevalier, C., Griffiths, R. R., Salvatori, T., Lukasiewicz, T., Berner, J. (2023). Mathematical capabilities of chatgpt. Advances in neural information processing systems, 36, 27699-27744

11. Elnashar, A., White, J., & Schmidt, D. C. Prompt engineering for structured data: a comparative evaluation of styles and LLM performance. Artificial Intelligence and Autonomous Systems, 2025, 2(2), 32-49. DOI: 10.55092/aias2025009

12. HE, Jia, et al. Does prompt formatting have any impact on llm performance?. arXiv preprint arXiv:2411.10541, 2024

O. Kis, M. Klymenko

ANALYSIS OF RISKS WHEN ENSURING INFORMATION SAFETY

The article presents a methodology for optimizing Large Language Models (LLMs) to process ultra-large contextual datasets for managerial decision support. This approach addresses the growing need for high-tech processing of unstructured information in digital management. The study examines the "lost-in-the-middle" problem and the resulting degradation in factual retrieval accuracy when input volume exceeds 100,000 tokens. Architectural limitations of transformers often lead to arithmetic hallucinations during complex calculations.

A hybrid concept is proposed, based on the integration of technical model auditing and Prompt Engineering techniques, including Dynamic Analytical Model Injection (DAP) and the transmission of precomputed statistical parameters (regression coefficients) in structured textual formats (JSON, Markdown, YAML, CSV). This ensures that the LLM operates on verified mathematical foundations rather than attempting to derive complex calculations from raw text alone.

A system of critical technical metrics (TTFT, TPOT, Needle-in-a-Haystack) is defined, which directly correlates with the responsiveness and validity of managerial decisions. The proposed structured data transmission methodology enables the minimization of arithmetic hallucinations and reduces the model's cognitive load. This approach ensures stable model performance and maintains analytical accuracy even when processing extra-large context volumes. The application of Prompt Engineering techniques allows the transformation of LLMs from text generation tools into full-fledged interpreters of complex analytical models without additional fine-tuning.

This hybrid approach increases enhances the transparency and reproducibility of analytical workflows within Decision Support Systems (DSS). By structuring quantitative parameters into machine-readable formats and aligning them with contextual evidence, the approach ensures a more reliable interpretation of diverse datasets. This minimizes analytical errors in large contexts and provides a reliable basis for strategic planning.

Keywords: Large Language Models (LLM), Prompt Engineering, RAG emulation, Decision Support Systems, managerial decision-making, analytical models, RAG, JSON, Markdown.

Надійшла до редакції: 10.02.2025
Прийнята до друку: 21.04.2026
Опубліковано: 27.04.2026