

АДАПТИВНІ RAG-СИСТЕМИ З АГЕНТНИМ УПРАВЛІННЯМ ЗНАННЯМИ ДЛЯ ЕВОЛЮЦІЇ ПРОГРАМНИХ ПРОДУКТІВ

Проценко М. М.

*здобувач вищої освіти третього (аспірантського рівня)
за спеціальністю 123 – Комп'ютерна інженерія*

Науковий керівник: Мірошник М. А.

доктор технічних наук, професор

*Міжнародний гуманітарний університет
м. Одеса, Україна*

У сучасному світі програмне забезпечення стає дедалі складнішим, а вимоги до нього постійно зростають. Проведене дослідження показує, що традиційні методи розробки та еволюції програмних продуктів часто не справляються з цими викликами. Великі мовні моделі (LLM) та технології доповненої генерації з використанням пошуку (RAG) створюють нові можливості для програмної інженерії, але навіть традиційні RAG-системи демонструють обмеження при обробці складних програмних артефактів. Інтеграція агентного управління знаннями в RAG-системи представляє інноваційний підхід, що створює адаптивні системи для еволюції програмних продуктів [1].

Традиційні RAG-системи, поєднуючи генеративні можливості LLM із механізмами пошуку в зовнішніх джерелах даних, показують покращення продуктивності на 27% порівняно з базовими LLM при тестуванні на бенчмарку BigCodeBench. Проте аналіз досліджень виявив, що при вирішенні багатоетапних задач рефакторингу точність таких систем знижується на 32%, а при роботі з великими кодовими базами час відгуку збільшується на 78% з падінням точності на 24%. Основними проблемами постають обмежена контекстуальна інтеграція та відсутність ітеративного міркування [2].

Для подолання цих обмежень у статті розглядаються адаптивні RAG-системи з агентним управлінням знаннями. Аналіз даних з набору HotpotQA показав, що такі системи перевершують традиційні на 34% при вирішенні завдань, що вимагають багатоетапного міркування. У ході дослідження було виявлено чотири ключові патерни, що забезпечують ефективність агентних RAG-систем. Рефлексія дозволяє агентам критично оцінювати власні рішення, виявляючи помилки. Дослідження системи Self-Refine показало зменшення синтаксичних помилок у згенерованому коді на 42%. Планування надає агентам можливість створювати структуровані

послідовності дій. Аналіз системи ReAct виявив скорочення часу планування рефакторингу на 51% при збереженні якості результату. Використання інструментів розширює можливості агентів через взаємодію із зовнішніми API та системами. Інтеграція з GitHub підвищила точність аналізу коду на 28%. Мультиагентна співпраця дозволяє спеціалізованим агентам розподіляти завдання між собою. Мультиагентні системи демонструють на 44% вищу точність при декомпозиції монолітних систем.

На основі аналізу літератури та дослідження існуючих рішень було виділено шість архітектурних варіантів агентних RAG-систем, кожен з яких оптимізований для різних аспектів еволюції програмного забезпечення. Одноагентна архітектура використовує єдиного інтелектуального агента, що ефективно для локалізованих задач. Мультиагентна архітектура розподіляє роботу між спеціалізованими агентами, що прискорює обробку різних аспектів кодової бази. Ієрархічна архітектура організовує агентів у багаторівневу структуру, де верхні рівні координують стратегічні рішення. Корективна архітектура впроваджує механізми самоперевірки та ітеративного покращення, що критично для завдань з високими вимогами до якості та безпеки. Адаптивна архітектура динамічно регулює стратегію роботи залежно від складності запиту, оптимізуючи використання ресурсів. Графова архітектура представляє знання у вигляді графа, що особливо ефективно для аналізу залежностей між компонентами системи [3].

У дослідженні було проаналізовано результати експериментів на бенчмарках BEIR та HotpotQA, які продемонстрували, що агентні RAG-системи значно покращують релевантність та точність відповідей порівняно з традиційними підходами. На HotpotQA агентні системи правильно відповідають на запитання в 75% випадків, тоді як традиційні моделі досягають лише 67% точності.

У різних аспектах еволюції програмних продуктів агентні RAG-системи демонструють вражаючі результати. У генерації коду системи з механізмом рефлексії створюють на 37% менше помилок та на 42% краще дотримуються архітектурних принципів. При виявленні дефектів агентні системи знаходять на 42% більше проблем та пропонують на 36% ефективніші виправлення. Аналіз даних показав, що в одному з досліджуваних проєктів мультиагентна система виявила 89% проблем продуктивності у 500,000 рядків коду [4].

Для рефакторингу архітектури ієрархічна агентна система досягла 88% точності при декомпозиції монолітної системи, що на 34% перевищує традиційні підходи. Дослідження показало, що така система успішно аналізує 2.3 млн рядків коду, пропонуючи плани, які зменшують зв'язність між компонентами на 47% та покращують когезію на 56%.

У документуванні коду агентні RAG-системи створюють на 61% точнішу документацію. Аналіз тестування на 100 API-методах показав

оцінку 4.3/5 за точність порівняно з 2.8/5 для традиційних підходів. У сфері архітектурних рішень агентні системи на 54% частіше рекомендують рішення, що відповідають довгостроковим вимогам проєкту.

Результати дослідження переконливо доводять, що адаптивні RAG-системи з агентним управлінням знаннями є потужним інструментом для прискорення еволюції програмних продуктів. Поєднуючи можливості LLM, механізми пошуку та агентне управління, ці системи ефективно вирішують складні задачі розробки, оптимізації та підтримки програмного забезпечення. Проаналізовані експериментальні дані демонструють значне покращення ефективності порівняно з традиційними підходами, що робить адаптивні RAG-системи перспективним напрямком для подальших досліджень та практичного застосування в індустрії програмного забезпечення.

Література:

1. Singh A., Ehtesham A., Kumar S., Talaei Khoei T. Agentic Retrieval-Augmented Generation: A Survey on Agentic RAG. arXiv Computer Science, 2025. URL: <https://arxiv.org/html/2501.09136v1> (дата звернення: 02.03.2025).
2. Wang L., Ma C., Feng X., Zhang Z. A Survey on Large Language Model Based Autonomous Agents. Frontiers of Computer Science, 2024. Vol. 18(6). URL: <https://arxiv.org/abs/2308.11432> (дата звернення: 01.03.2025).
3. Jeong S., Baek J., Cho S., Hwang S.J., Park J.C. Adaptive-RAG: Learning to Adapt Retrieval-Augmented Large Language Models through Question Complexity. arXiv Computer Science, 2024. URL: <https://arxiv.org/html/2403.14403v1> (дата звернення: 09.03.2025).
4. Yan S-Q., Gu J-C., Zhu Y., Ling Z-H. Corrective Retrieval Augmented Generation. arXiv Computer Science, 2024. URL: <https://arxiv.org/html/2401.15884v1> (дата звернення: 04.03.2025).

DOI <https://doi.org/10.36059/978-966-397-491-0-33>

КІБЕРБЕЗПЕКА ЯК СКЛАДОВА НАЦІОНАЛЬНОЇ БЕЗПЕКИ

Прядка С. А.

магістр

Міжнародний гуманітарний університет

м. Одеса, Україна

Кібербезпека є ключовим елементом національної безпеки України. У 2015 році було ухвалено Стратегію національної безпеки, яка визначає