УДК 004.65

ОПТИМІЗАЦІЯ ВЗАЄМОДІЇ З НЕЙРОННИМИ МЕРЕЖАМИ ЧЕРЕЗ ІНТЕГРАЦІЮ RAG І LLM

В. А. Поліщук, І. З. Миклушка

Українська академія друкарства,
вул. Під Голоском, 19, Львів, 79020, Україна

Запропоновано метод управління моделями штучного інтелекту, що базується на використанні RAG (Retrieval-Augmented Generation) та LLM як центрального медіатора, для оптимізації процесу взаємодії користувача з іншими нейронними мережами. Такий підхід передбачає інтеграцію сучасних технологій обробки лінгвістичного контенту з алгоритмами пошуку, що дозволяє здійснювати точне та ефективне впорядкування відповідних даних для генерації текстів або ж будь-якого іншого контенту, згенерованого штучним інтелектом. Дослідження охоплює статус сучасних моделей з архітектурою RAG, їхню роль і місце на ринку IT-сервісів. Додатково розглянуто, власне, технічні перспективи застосування RAG у співпраці з LLM, такими як GPT. Запропонований архітектурний концепт дозволяє системам не тільки підвищувати точність інформації, але й адаптуватися до нових вимог і завдань, спрощуючи взаємодію між людиною і штучним інтелектом. Крім того, коротко описано технічні аспекти використання RAG і LLM, зокрема контекстний пошук та вплив цих технологій на персоналізацію відповідей і спеціалізацію систем на вузькоспеціалізовані завдання, що сприяє поліпшенню їхніх функціональних можливостей та ефективності у комерційних умовах.

Ключові слова: *Нейронні мережі, великі мовні моделі (LLM), медіатор між моделями, оптимізація взаємодії, контекстний пошук, ефективність ШІ-систем, модульна інтеграція, генерація контенту, штучний інтелект (ШІ), адаптивні системи, автоматизація процесів, Retrieval-Augmented Generation (RAG), персоналізація відповідей, інтеграція технологій.*

Постановка проблеми. Сучасний світ технологій та штучного інтелекту (ШІ) постійно розвивається, пропонуючи дедалі більш вдосконалені та інтелектуальні рішення для різноманітних завдань та сервісів. Широке застосування ШІ в різних сферах життя відкриває нові можливості для автоматизації процесів і водночас ставить перед розробниками нові виклики. Однією з основних проблем сучасних сервісів є те, що більшість із них базується на використанні окремих, часто ізольованих один від одного моделей штучного інтелекту, що обмежує можливість глибокої взаємодії та інтеграції різноманітних інтелектуальних систем, а також ускладнює обробку складних мультидисциплінарних запитів, які вимагають комплексного підходу. У цьому контексті поєднання Retrieval-Augmented Generation (RAG) методу

з великими мовними моделями (LLM) та іншими нейронними мережами відкриває перед розробниками нові перспективи. RAG, що ефективно використовує можливості пошуку та генерації відповідей на основі великих обсягів даних, дозволяє створювати системи, які не лише мають глибокі знання в певній галузі, але й здатні ефективно використовувати інформацію з різних джерел для розв'язання комплексних завдань. Це стає можливим завдяки тому, що RAG і LLM-модель можуть слугувати медіатором між користувачем та різноманітними ШІ-системами, об'єднуючи їхні зусилля для забезпечення більш точних, гнучких та адаптивних рішень.

Аналіз останніх досліджень та публікацій. Аналіз останніх досліджень показує, що Retrieval-Augmented Generation (RAG) є ключовою технологією у сфері обробки природної мови. За останні роки з'явилося багато досліджень, які демонструють значний прогрес у цій галузі. Одна з головних переваг RAG — це можливість подолати недоліки великих мовних моделей (LLM), такі як генерація неправдивої чи неадекватної інформації та обмеження під час тренування [1]. Використовуючи зовнішні джерела даних, RAG покращує точність і надійність інформації, що надається моделлю. Крім того, дослідження вказують на успішну інтеграцію RAG з великими мовними моделями для створення більш гнучких і масштабованих систем. Ці системи використовують векторні бази даних, що дає змогу ефективно взаємодіяти з базами даних і швидше обробляти запити.

Варто також виокремити зростаючу популярність RAG у комерційних застосунках. Наприклад, компанія Databricks [2] активно впроваджує ці моделі у свої рішення для покращення спостережуваності даних, забезпечуючи більш надійну та релевантну інформацію для прийняття рішень.

Мета статті — дослідження потенціалу та перспектив використання RAG-архітектури у поєднанні з великими мовними моделями (LLM) для інтеграції різних моделей штучного інтелекту в єдину систему. Основна увага приділяється аналізу переваг такої інтеграції, а також можливостям створення багатофункціональних систем управління та сервісів.

Виклад основного матеріалу дослідження. Архітектура RAG (Retrieval-Augmented Generation) та модель LLM (Large Language Models) є передовими методами в галузі обробки природної мови (natural language), що використовуються для підвищення точності та релевантності автоматично генерованих відповідей. RAG покращує процес генерації відповідей шляхом вилучення даних із зовнішніх джерел перед їхньою обробкою, забезпечуючи актуальність інформації, використовуваної для відповідей. LLM, як GPT, здатні генерувати текст, що імітує людську мову, з високою точністю та природністю, спираючись на великі масиви навчальних даних.

Комбінація цих двох підходів дозволяє системі не тільки забезпечувати високу точність відповідей, але й ефективно взаємодіяти з іншими моделями ШІ, адаптуватися до нових вимог і даних та вдосконалювати «спілкування» між людиною та машиною. Така система може ефективно слугувати медіатором у взаємодії між людьми та складними аналітичними інструментами, полегшуючи доступ до

інтерактивних та інтелектуальних систем, що сприяє розвитку автоматизованих систем відповідей та комплексних систем рішень.

Одна з причин, чому саме LLM з RAG є чудовим кандидатом для того, щоб бути так званим «медіатором» у складніших системах ШІ, – це масштабованість та адаптивність. Такі системи можуть швидко адаптуватися до нових даних і змінених вимог користувачів, що дозволяє їм залишатися корисними та ефективними в різних ситуаціях. Використання великих масивів даних дозволяє LLM оновлювати свої відповіді з новою інформацією, тоді як RAG забезпечує точне вилучення та використання цієї інформації для покращення відповідей [1]. Здатність швидко пристосовуватися підвищує якість сервісу і дозволяє системі зростати та інтегрувати нові модулі без необхідності повного перепроєктування або значних змін у основній архітектурі [2].

Практичність та відповідність до інженерних стандартів зробили RAG доволі популярним рішенням при створенні сервісів з використанням LLM. Однак існують важливі завдання, які ще потрібно вирішити:

– покращення контекстного пошуку;
– підвищення точності вибору документів у великих базах даних та забезпечення безпеки інформації, наприклад, уникнення ненавмисного розкриття джерел або метаданих великими мовними моделями (LLM) [3].

Розвиток екосистеми RAG тісно пов›язаний з удосконаленням її технічних засад. Такі платформи, як LLamaIndex та LangChain, стали дуже популярними після появи ChatGPT, надаючи спеціалізовані RAG-API і ставши незамінними в галузі LLM. Водночас нові технологічні рішення можуть бути менш функціональними, проте відрізнятися своєю орієнтованістю на спеціальні та специфічні завдання. Популярні постачальники програмного забезпечення та хмарні сервіси також розширюють свої пропозиції, вміщуючи послуги, орієнтовані на RAG. Наприклад, Verba 11 від Weaviate використовується для внутрішніх завдань компаній [4], тоді як Kendra від Amazon пропонує розумний пошук для корпоративних потреб, дозволяючи переглядати контент через інтегровані з›єднання.

У розвитку технології RAG спостерігаються три головні напрямки: персоналізація, спрощення та спеціалізація на вузькоспрямовані завдання. Загалом усі ці напрямки можна утотожнити – як полегшення використання для зниження вхідного порогу та оптимізація для ефективнішої роботи в промислових умовах. Спільний прогрес моделей RAG та їхніх технологічних стеків є очевидним, а технологічні інновації неперервно підіймають планку для сучасної інфраструктури. Завдяки вдосконаленням у технологічних стеках можливості RAG невпинно розширюються.

Одне з головних питань при спробі поєднати LLM, RAG та інші моделі штучного інтелекту – це, власне, сам спосіб спілкування між моделями. Використання моделі LLM спільно з RAG як медіатора дозволяє створити ефективну систему, де кожен окремий AI-сервіс комунікує виключно через самоізольовані API. Цей підхід забезпечує централізоване управління взаємодією між різними сервісами, де медіатор (LLM + RAG) виступає як єдиний контактний пункт для всіх запитів та відповідей.

Така організація спілкування мінімізує залежності між окремими компонентами системи, підвищує її масштабованість та полегшує подальше управління.

Цей підхід має свій аналог у класичному програмуванні, зокрема у використанні поведінкового патерну «Mediator” («Посередник»). У цьому контексті медіатор управляє взаємодією між різними об›єктами системи, роблячи їх менш залежними один від одного та спрощуючи їхню взаємодію. Також можна згадати патерн «Команди», який використовується для інкапсуляції всіх деталей операції в окремому об’єкті, що дозволяє параметризувати об›єкти за допомогою конкретних дій та легко змінювати логіку виконання команд у медіаторі без втручання в код, який використовують інші компоненти системи. Це сприяє створенню гнучких та легко модифікованих систем, де можлива швидка адаптація до нових вимог без необхідності зміни основної архітектури системи.

Схема, представлена на рис. 1, зображає зв’язки між окремими сервісами та централізованим медіатором. За допомогою цих інтерфейсів, що забезпечують ізольоване спілкування LMM та окремих екземплярів ШІ, кожна нейронна мережа може обмінюватися даними та запитами з медіатором, оптимізуючи таким способом обробку та розподіл обов’язків.

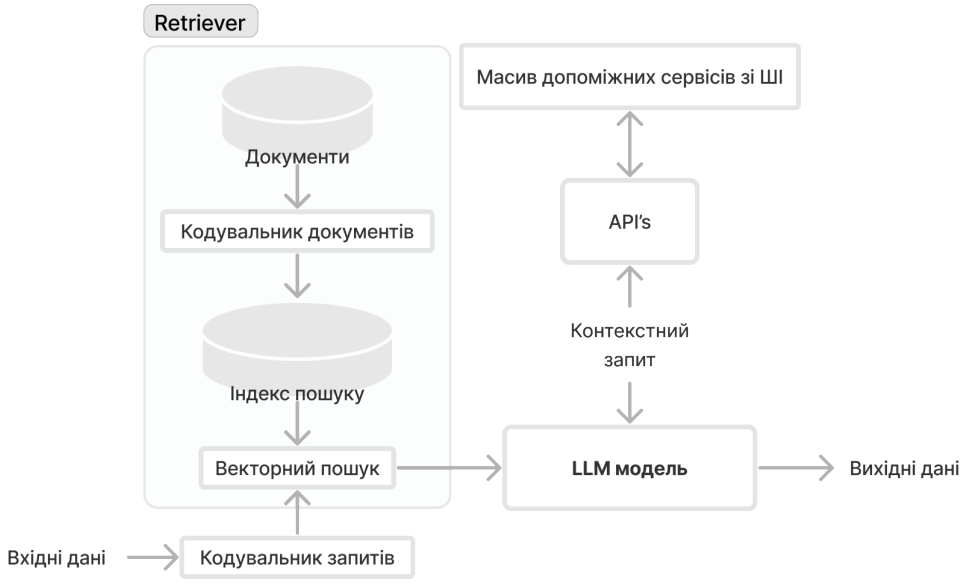

Рис. 1. Схема взаємодії RAG, LLM та інших сторонніх сервісів зі ШІ

Розглянемо детальніше процеси, представлені на рис. 1. На схемі зображено процес взаємодії між ретривером (Retriever), масивом допоміжних сервісів із штучного інтелекту та моделлю LLM. Процес розпочинається із вхідних даних, які подаються на енкодер запитів (Query encoder), де вони перетворюються на відповідні векторні представлення для пошуку. Тим часом у ретривері знаходяться документи, які обробляються енкодером документів (Document encoder) і розміщуються в індексі пошуку, оптимізованому для швидкого векторного пошуку. Під час

векторного пошуку система визначає найбільш релевантні документи для даного запиту, які далі передаються до моделі LLM. В той самий час масив допоміжних сервісів з ШІ через окремі API обмінюється даними та запитами з медіатором LLM, що забезпечує вибірку необхідної інформації для синтезу вихідних даних. Модель LLM інтегрує отримані відомості та формує відповідь, яка потім подається як кінцевий результат користувачу. Слід зазначити, що кінцевий контент, який отримає клієнт, може бути довільним – починаючи від текстової відповіді й закінчуючи комбінацією тексту, зображень, аудіо тощо.

Інтеграція модулів через сторонні API у архітектурі, що базується на LLM, відкриває значні перспективи для розширення функціональності та масштабованості системи. До прикладу, GPT-4 може ефективно взаємодіяти з різноманітними сторонніми сервісами, отримуючи або надсилаючи дані через їх API, що дозволяє створювати гнучкі рішення, які можуть адаптуватися до специфічних потреб користувачів. Завдяки такому підходу кожен сервіс, чи то для пошуку даних, перекладу, обробки зображень, чи для інших завдань, може бути легко інтегрований у загальний механізм, забезпечуючи довільний персоналізований функціонал в межах однієї системи. Така модель, як GPT-4, обробляє вхідні запити, використовуючи надану інформацію від ретривера та інших сервісів, інтегруючи їх для синтезу кінцевої відповіді. Наявність API як містка для взаємодії між різними системами і компонентами дозволяє моделі GPT-4 стати центральним вузлом у генеруванні відповідей, які вимагають спеціалізованих знань або функціональних можливостей. Інтеграція через API уможливлює плавне з'єднання внутрішньої логіки GPT-4 із зовнішніми джерелами та сервісами, причому спілкування між сервісами та центральним медіатором може проводитись навіть за допомогою text-to-text, text-to-audio, text-to-image запитів, згенерованих самим медіатором. На цьому етапі можуть виникати труднощі зі стабільністю та строгістю запитів, адже текстові запити, згенеровані LLM-моделлю, повинні мати контекстне обґрунтування та бути достатньо стабільними, щоб API та інші сторонні сервіси повертали бажаний результат. До проблемних місць такої архітектури також потрібно враховувати і самі проблеми RAG-підходу, оскільки існує багато нерозв'язаних проблем з контекстним пошуком та критеріями вибору того чи іншого варіанту як «правильного» чи «відповідного» згідно з контекстом запиту.

Висновки. Отже, ми дослідили можливості архітектури, побудованої на основі поєднання RAG-підходу та LLM-моделі, а також «спілкування» цієї архітектури з іншими сторонніми сервісами, які теж використовують ШІ. Розкрито основні переваги та недоліки такого поєднання, а також коротко описані перспективи використання RAG для специфічних завдань.

Архітектура RAG з інтеграцією таких великих мовних моделей, як LLM, є значним кроком у розвитку штучного інтелекту, що забезпечує високу точність та релевантність у генерації автоматичних відповідей та підвищенні інтелектуального потенціалу систем. Використання цих технологій сприяє створенню складних багатофункціональних систем, які можуть слугувати ефективними медіаторами

між користувачем та різноманітними AI-системами. Наявність централізованого управління через API забезпечує плавну інтеграцію і сприяє зниженню вхідного порогу для користувачів та розробників, даючи можливість модульного під›єднання сервісів до центрального медіатора. Водночас платформи, що використовують RAG, демонструють не лише можливість швидкої адаптації до нових даних, але й стійкість до змін у вимогах, сприяючи оптимізації роботи в промислово-комерційних умовах. Перспективи розвитку технології RAG зумовлені її здатністю до масштабування, а також адаптивністю та спеціалізацією на вузькоспрямовані завдання, що в кінцевому підсумку сприяє посиленню функціональних можливостей та зростанню ефективності інтелектуальних систем.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, Haofen Wang. Retrieval-Augmented Generation for Large Language Models: A Survey. Feb 27, 2024. 2−3 pp.
2. Kyle Kirwan. The Rise of RAG-Based LLMs in 2024. January 15, 2024. 2 pp.
3. Demiao Lin. Revolutionizing Retrieval-Augmented Generation with Enhanced PDF Structure Recognition. 23 Jan, 2024. 1−2 pp.
4. Yunfan Gao. Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, Haofen Wang. Retrieval-Augmented Generation for Large Language Models: A Survey. 27 Mar, 2024. 15 pp.
5. Alon U., Xu F., He J., Sengupta S., Roth D., Neubig G. Neurosymbolic language modeling with automaton-augmented retrieval, in International Conference on Machine Learning. PMLR, 2022. 468–485 pp.
6. An Open-Source Tool to Build Your Own RAG Retrieval Augmented Generation Pipeline and Utilize LLMs for Internal-Based Outputs. URL:
7. https://www.marktechpost.com/2023/09/11/meet-verba-an-open-source-tool-to-build-your-own-rag-retrieval-augmented-generation-pipeline-and-utilize-llms-for-internal-based-outputs/.

REFERENCES

1. Yunfan, Gao, Yun, Xiong, Xinyu, Gao, Kangxiang, Jia, Jinliu, Pan, Yuxi, Bi, Yi, Dai, Jiawei, Sun, Qianyu, Guo, Meng, Wang, & Haofen, Wang. (Feb 27, 2024). Retrieval-Augmented Generation for Large Language Models: A Survey. 2−3 (in English).
2. Kyle, Kirwan. (January 15, 2024). The Rise of RAG-Based LLMs in 2024. 2 (in English).
3. Demiao, Lin. (23 Jan 2024). Revolutionizing Retrieval-Augmented Generation with Enhanced PDF Structure Recognition (in English).
4. Yunfan, Gao, Yun, Xiong, Xinyu, Gao, Kangxiang, Jia, Jinliu, Pan, Yuxi, Bi, Yi, Dai, Jiawei, Sun, Qianyu, Guo, Meng, Wang, & Haofen, Wang. (27 Mar, 2024). Retrieval-Augmented Generation for Large Language Models: A Survey (in English).
5. Alon, U., Xu, F., He, J., Sengupta, S., Roth, D., & Neubig, G. (2022). Neurosymbolic language modeling with automaton-augmented retrieval, in International Conference on Machine Learning. PMLR (in English).

6. An Open-Source Tool to Build Your Own RAG Retrieval Augmented Generation Pipeline and Utilize LLMs for Internal-Based Outputs. Retrieved from
7. https://www.marktechpost.com/2023/09/11/meet-verba-an-open-source-tool-to-build-your-own-rag-retrieval-augmented-generation-pipeline-and-utilize-llms-for-internal-based-outputs/ (in English).

doi: 10.32403/1998-6912-2024-1-68-46-53

OPTIMIZATION OF INTERACTION WITH NEURAL NETWORKS THROUGH RAG AND LLM INTEGRATION

V. A. Polishchuk, I. Z. Myklushka

Ukrainian Academy of Printing,
19, Pid Holoskom St., Lviv, 79020, Ukraine
valentinepolischuk2@gmail.com,
myklushka@gmail.com

This article introduces a method for managing artificial intelligence (AI) models by leveraging Retrieval-Augmented Generation (RAG) as a central mediator. This approach optimizes the interaction between users and various neural networks, utilizing advanced linguistic processing technologies integrated with sophisticated search algorithms. This synergy ensures efficient organization and retrieval of relevant data, facilitating the generation of textual and other forms of content by AI systems. The methodology enhances the interface between AI systems and users, making it more adaptable and user-friendly for complex task management. By incorporating RAG, these systems can effectively utilize existing data, allowing for dynamic adaptations to changing conditions in real time. This not only improves the quality of interactions but also boosts the efficiency of processing requests, thereby expanding the potential for both creative and analytical applications across various fields.

The article also explores the potential of combining RAG with Large Language Models (LLMs) like GPT. This combination aims to develop multifunctional AI systems that enhance accuracy and adaptability, facilitating smoother interactions between humans and machines. The integration of RAG with LLMs promises to refine the responsiveness of AI systems to user inputs, making them more intuitive and capable of handling nuanced interactions. The text elaborates on the scalability and adaptability of RAG technologies, which are critical for the development and future evolution of intelligent systems. Using centralized management through APIs, RAG-enabled systems support modular integration of services, which simplifies the incorporation of new functionalities and enhances the overall system flexibility. This approach reduces the entry threshold for users and developers, fostering an environment where continuous improvement and customization are feasible. The deployment of RAG in AI systems exemplifies a significant advancement in AI

technology, emphasizing its potential to transform the landscape of AI interactions and system design.

Keywords: *neural networks, large language models (LLM), mediator between models, optimization of interaction, contextual search, efficiency of AI systems, modular integration, content generation, artificial intelligence (AI), adaptive systems, process automation, Retrieval-Augmented Generation (RAG), response personalization, technology integration.*

Стаття надійшла до редакції 16.05.2024.
Received 16.05.2024.