УДК 004.8

Н. В. Танасійчук
В. О. Денисюк

ВИКОРИСТАННЯ RAG ДЛЯ ПОКРАЩЕННЯ ВІДПОВІДЕЙ У ВЕЛИКИХ МОВНИХ МОДЕЛЯХ

Вінницький національний технічний університет

Анотація

У роботі розглядається підхід Retrieval-Augmented Generation (RAG) як ефективний метод підвищення точності, релевантності та надійності відповідей великих мовних моделей. Описано принцип інтеграції зовнішніх джерел знань у процес генерування відповідей, що дозволяє моделі опрацьовувати актуальну інформацію та зменшувати ризик галюцинацій. Проаналізовано ключові компоненти RAG-архітектури, механізми індексування та пошуку, а також переваги порівняно з традиційними LLM, які працюють виключно на основі попереднього навчання.

Ключові слова: *Retrieval-Augmented Generation, великі мовні моделі, підвищення точності відповідей.*

Abstract

This paper examines the Retrieval-Augmented Generation (RAG) approach as an effective method for improving the accuracy, relevance, and reliability of large language model responses. It describes the principle of integrating external knowledge sources into the response generation process, which allows the model to process relevant information and reduce the risk of hallucinations. The key components of the RAG architecture, indexing and search mechanisms, and advantages over traditional LLMs that operate solely on the basis of prior training are analyzed.

Keywords: *Retrieval-Augmented Generation, large language models, improving answer accuracy.*

Вступ

Стрімкий розвиток великих мовних моделей (LLM) відкрив широкі можливості для автоматизації аналізу текстових даних, створення інтелектуальних систем та підтримки користувачів у різних галузях. Водночас зростає потреба у підвищенні точності, обґрунтованості та актуальності відповідей, оскільки моделі, засновані лише на попередньому навчанні, мають обмежений доступ до нових даних і можуть генерувати некоректні або застарілі твердження. Це створює виклик для побудови систем, здатних поєднувати потужність генеративних моделей із достовірними зовнішніми джерелами знань.

Підхід Retrieval-Augmented Generation (RAG) пропонує вирішення цієї проблеми шляхом інтеграції механізмів пошуку інформації безпосередньо в процес генерації відповідей [1]. Використання RAG дозволяє моделі динамічно звертатися до актуальних даних, забезпечувати кращу аргументованість результатів та знижувати ризик появи галюцинацій. У цьому дослідженні розглядаються принципи роботи RAG, його архітектура та переваги для побудови сучасних інтелектуальних систем.

Основна частина

Використання великих мовних моделей (LLM) у сучасних інформаційних системах відкриває можливості для автоматизації аналітики, генерації текстів та підтримки прийняття рішень. Однак значна проблема, притаманна таким моделям, це явище галюцинацій. Під галюцинаціями мовних моделей розуміють ситуації, коли модель генерує неправдиву, вигадану або логічно некоректну інформацію, що не ґрунтується на реальних даних[2]. Це може проявлятися у вигаданих фактах, неточних посиланнях, неправильних поясненнях або надмірній узагальненості. Причини галюцинацій полягають у статистичній природі LLM - вони генерують текст на основі ймовірнісних залежностей у навчальних даних, а не на основі перевірених знань. Крім того, моделі не мають вбудованого доступу до актуальних даних, тому не можуть коригувати свою інформацію на основі реальних джерел.

Для подолання цієї проблеми запропоновано підхід Retrieval-Augmented Generation (RAG), що інтегрує механізми пошуку інформації у процес генерації відповідей. На відміну від звичайних LLM, які покладаються виключно на внутрішні параметри, RAG дозволяє звертатися до зовнішніх баз знань, документів або індексованих сховищ даних. Це створює можливість отримувати актуальні, підтверджені та контекстно релевантні відомості навіть у випадках, коли інформація відсутня у навчальному корпусі моделі. Загалом RAG знижує ризик галюцинацій завдяки тому, що відповідь

моделі будується не тільки на основі мовної генерації, а й на основі реальних джерел, які попередньо пройшли пошук за релевантністю.

Також використання RAG сприяє покращенню відтворюваності результатів.

У традиційних LLM відповіді можуть істотно змінюватися залежно від формулювання запиту, що ускладнює їх застосування в задачах, де потрібна стабільність. У RAG значна частина змісту відповіді залежить від знайдених документів, тому модель виробляє більш послідовні та передбачувані результати. Це особливо важливо у прикладних системах, що працюють з нормативними актами, технічною документацією або науковими матеріалами, де непослідовність може бути критичною.

Архітектура RAG складається з двох основних компонентів: модуля пошуку (retriever) та модуля генерації (generator) [3].

Модуль retriever відповідає за індексування колекції документів та пошук релевантних фрагментів на основі запиту користувача. Як правило, тексти попередньо кодуються у векторні представлення за допомогою моделей для семантичного пошуку. Після цього система обчислює схожість між вектором запиту та векторами документів, обираючи найбільш відповідні фрагменти.

Далі модуль генерації отримує знайдені документи разом із запитом користувача та формує остаточну відповідь, яка використовує як контекст, так і отримані дані. У такий спосіб генерація базується не лише на параметрах мовної моделі, а й на фактичній інформації з бази знань.

Ефективність RAG проявляється у кількох аспектах.

По-перше, система забезпечує високу актуальність відповідей, оскільки джерелом інформації є динамічно оновлювані сховища. Це особливо важливо для доменів, де дані швидко змінюються: фінанси, медицина, правові документи, технічні специфікації.

По-друге, покращується точність і обґрунтованість, оскільки модель відштовхується від конкретних текстових фрагментів, а не генерує інформацію на основі припущень.

По-третє, RAG дозволяє будувати системи, що відповідають вимогам прозорості: користувач або розробник можуть переглянути, які саме документи стали основою для відповіді. Це важливо у задачах, де необхідна пояснюваність, зокрема в експертних системах.

Окрему увагу слід звернути на прикладне значення використання RAG у сучасних системах [4]. Зокрема, у діловій аналітиці RAG може забезпечувати доступ до великих масивів внутрішніх документів компанії, оперативно знаходячи потрібні дані та виключаючи ризик генерації недостовірної інформації.

В освітніх платформах RAG може формувати пояснення, що базуються на реальних навчальних матеріалах, не створюючи нових або помилкових тверджень.

У правовій галузі цей підхід дозволяє моделі працювати з нормативно-правовими актами, забезпечуючи точність і відповідність чинним документам.

Завдяки цьому RAG виступає універсальним інструментом для побудови систем, де критично важливо зберігати достовірність, контрольованість і пояснюваність результатів роботи великих мовних моделей.

Висновки

Запровадження Retrieval-Augmented Generation створює новий підхід до розробки інтелектуальних систем, які поєднують гнучкість генеративних моделей із надійністю зовнішніх джерел знань. Значно зменшуючи кількість помилок та некоректних тверджень, RAG стає фундаментальним інструментом для побудови систем зі стійкою якістю відповідей.

Використання RAG дає змогу розширити можливості LLM, забезпечити контроль над інформаційною базою та сформувати більш надійні рішення для наукових, освітніх, комерційних та технічних застосувань.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Semyvolos L. How retrieval augmented generation (RAG) makes LLM smarter. AltexSoft. URL: https://www.altexsoft.com/blog/retrieval-augmented-generation-rag/.
2. Що таке галюцинації LLM? Причини, етичні проблеми та профілактика. UNITE AI. URL: https://www.unite.ai/uk/what-are-llm-hallucinations-causes-ethical-concern-prevention/.
3. Retrieval augmented generation (RAG) and semantic search for gpts. OpenAI. URL: https://help.openai.com/en/articles/8868588-retrieval-augmented-generation-rag-and-semantic-search-for-gpts .
4. RAG: доповнена генерація з пошуком для точних ШІ-відповідей. blog.colobridge.net. URL: https://blog.colobridge.net/uk/2025/11/search-augmented-generation-rag-ua .

Танасійчук Назар Володимирович – студент групи 2КН-24м, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, email: minex.sch@gmail.com

Денисюк Валерій Олександрович – канд. техн. наук, доцент, доцент кафедри комп'ютерних наук, Вінницький національний технічний університет, м.Вінниця, e-mail: vad64@i.ua.

Tanasiychuk Nazar Volodymyrovych - student of Computer Science Department, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: minex.sch@gmail.com

Denysiuk Valerii Olexandrovich – Ph.D., Assistant Professor, Assistant Professor of the Chair of Computer Science, Vinnytsia National Technical University, Vinnytsia, e-mail: vad64@i.ua