

АРХІТЕКТУРА СЕРВЕРНОЇ СИСТЕМИ ШІ-ПОМІЧНИКА НА ОСНОВІ RAG-ПІДХОДУ ТА ВЕКТОРНОГО ПОШУКУ

Вінницький національний технічний університет

Анотація

У роботі представлено архітектуру спеціалізованої інформаційної системи ШІ-помічника, що реалізує підхід Retrieval-Augmented Generation (RAG) для роботи з великими масивами текстових документів. Спеціалізований ШІ-помічник використовує векторний пошук для надання точних та контекстних відповідей на запитання користувачів у межах визначеної предметної області. Реалізовано серверну програмну частину (бекенд-сервіс) із використанням сучасних технологій: Python, FastAPI, PostgreSQL із розширенням pgvector. Система інтегрує модуль семантичної індексації документів, спеціалізовану векторну базу знань, механізм інтелектуального пошуку та мовну модель для генерації відповідей. Особливу увагу приділено побудові надійного конвеєра обробки даних (ETL), проєктуванню структури бази знань та забезпеченню масштабованості й розширюваності бекенд-архітектури. У результаті створено зручний у інтеграції сервіс, який надає інтерфейси програмного доступу (API) для побудови діалогових систем та галузевих інформаційних рішень на основі ШІ.

Ключові слова: інформаційна система, спеціалізований ШІ-помічник, RAG, векторний пошук, база знань, бекенд-сервіс, Python, FastAPI, pgvector, семантичний пошук.

Abstract

The paper presents the architecture of a specialized AI-assistant information system that implements the Retrieval-Augmented Generation (RAG) approach for working with large collections of textual documents. The specialized AI assistant employs vector search to provide accurate and context-aware responses to user queries within a defined subject domain. The server-side component (backend service) is implemented using modern technologies, including Python, FastAPI, and PostgreSQL with the pgvector extension. The system integrates a module for semantic document indexing, a specialized vector-based knowledge repository, an intelligent retrieval mechanism, and a language model for answer generation. Particular attention is paid to the development of a reliable data-processing pipeline (ETL), the design of the knowledge base structure, and ensuring the scalability and extensibility of the backend architecture. As a result, a highly integrable service has been developed, offering programmatic access interfaces (API) for building dialogue systems and domain-specific AI-driven information solutions.

Keywords: information system, specialized AI assistant, RAG, vector search, knowledge base, backend service, Python, FastAPI, pgvector, semantic search.

Вступ

У сучасних умовах швидкого зростання обсягів цифрових даних користувачі все частіше працюють із вузькоспеціалізованою інформацією: технічною документацією, внутрішніми регламентами, галузевими стандартами, навчально-методичними матеріалами. Навіть за наявності електронних архівів та пошуку за ключовими словами пошук потрібних відомостей у великих масивах тексту залишається складним і трудомістким завданням. Класичні підходи часто не враховують контекст, термінологічні особливості та структуру предметної області, що призводить до отримання нерелевантних або надто загальних результатів.

Розвиток технологій штучного інтелекту, зокрема підходу Retrieval-Augmented Generation (RAG) та векторного пошуку, відкриває можливість створення спеціалізованих ШІ-помічників, здатних поєднувати семантичний пошук із генерацією відповідей природною мовою на основі реальних документів. Побудова такого рішення у вигляді окремого бекенд-сервісу дає змогу інтегрувати його в існуючі інформаційні системи, корпоративні платформи чи навчальні середовища. Це дозволяє зменшити час пошуку інформації, підвищити точність відповідей та забезпечити зручну взаємодію користувача із предметною базою знань через уніфіковані програмні інтерфейси.

Метою цієї роботи є розроблення архітектури спеціалізованого ШІ-помічника, який використовує семантичне індексування, векторні представлення тексту та RAG-процедури для точного й обґрунтованого пошуку інформації в межах предметної області.

Результати дослідження

Інформаційна система спеціалізованого ШІ-помічника на основі RAG розроблена для вирішення проблеми швидкого та збалансованого отримання точних відповідей на запитання користувачів у межах визначеної предметної області. Основна мета системи – надати можливість інтерактивної роботи з базою знань шляхом поєднання векторного пошуку та генерації відповідей природною мовою на основі реальних документів і

структурованих даних [1].

Система призначена для:

- співробітників організацій, які працюють з великою кількістю внутрішньої документації;
- фахівців служби підтримки, яким потрібен швидкий доступ до актуальної інформації для відповіді на запити користувачів;
- аналітиків та менеджерів, що потребують узагальнених відповідей на основі кількох джерел даних;
- нових співробітників, які проходять процес онбордингу і потребують пояснень щодо процесів, сервісів та внутрішніх правил;
- студентів і викладачів, які працюють з навчально-методичними матеріалами певної спеціалізації

Випадки використання інформаційної системи:

- Реєстрація та автентифікація користувачів: користувач створює обліковий запис або входить у існуючий профіль для збереження історії запитів, налаштувань та обраних розділів бази знань.
- Формування та оновлення бази знань: уповноважений користувач (адміністратор або контент-менеджер) додає чи оновлює документи предметної області (інструкції, регламенти, технічні описи, навчальні матеріали), які зберігаються в системі та готуються до семантичної індексації.
- Обробка запиту користувача: користувач формулює запит природною мовою; система виконує векторний пошук релевантних фрагментів у базі знань, формує контекст і передає його до мовної моделі для побудови відповіді.

На рисунку 1 подано структурну схему системи, яка відображає послідовність обробки даних – від первинного документа до відповіді, згенерованої ШІ-помічником на основі релевантних фрагментів.

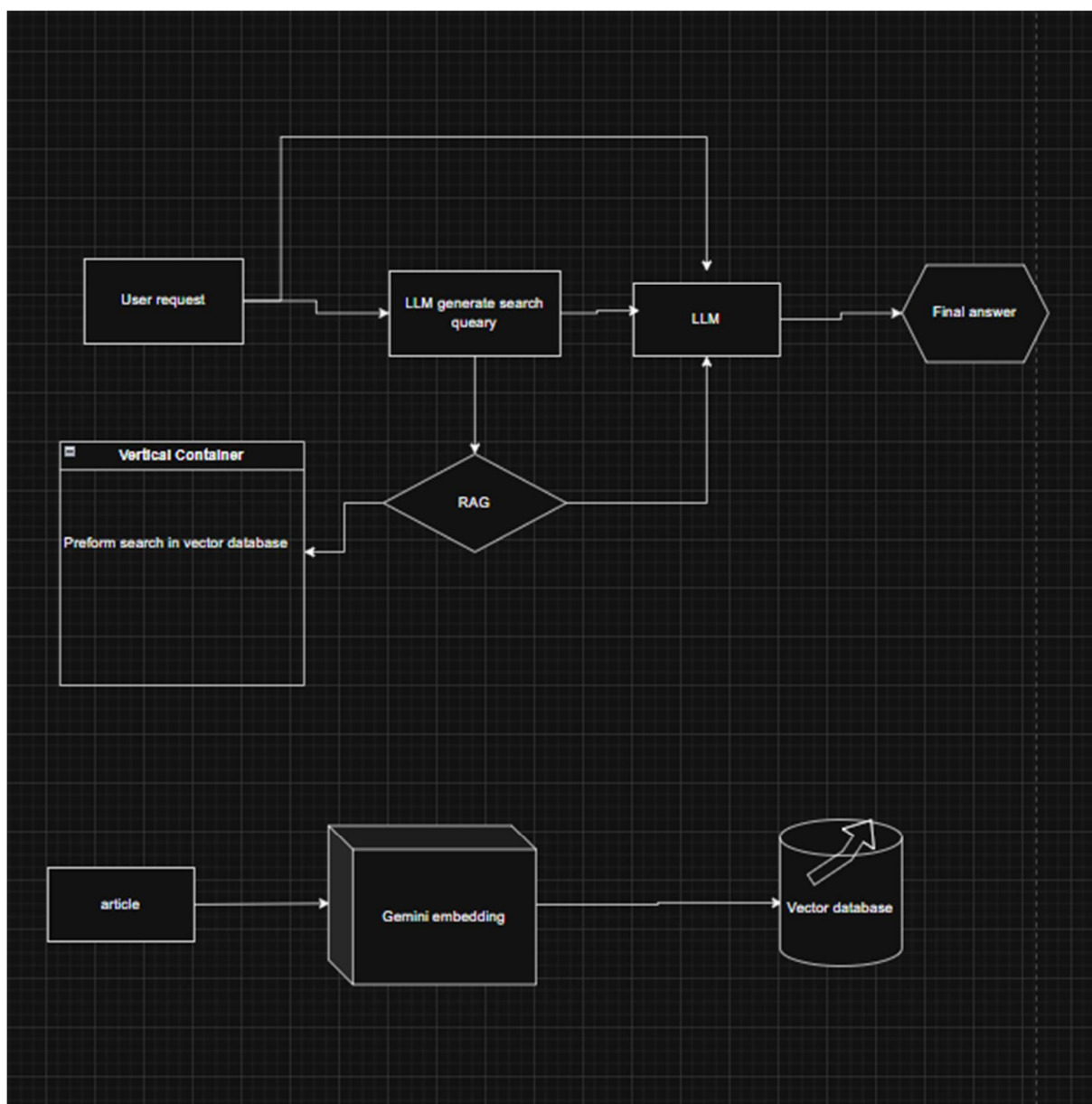


Рисунок 1 – Структурна схема системи

Опис процесу наступний:

1. Підготовка документації: вихідні текстові матеріали (документація, інструкції, статті, внутрішні регламенти) імпортуються в інформаційну систему та проходять попередню обробку (парсинг, очищення, нормалізацію).

2. Розбиття на фрагменти: оброблена документація ділиться на логічні фрагменти (абзаци, розділи, блоки знань), які можуть бути використані як окремі одиниці пошуку.

3. Формування векторних подань: для кожного фрагмента документації обчислюються векторні представлення за допомогою модуля Gemini embedding.

4. Запис у векторну базу даних: отримані вектори разом із посиланнями на вихідні фрагменти зберігаються у векторній базі даних (Vector database pgvector [3]).

5. Побудова пошукового запиту: на основі реального запиту користувача (User request) мовна модель (LLM generate search query) формує уточнений пошуковий запит, придатний для векторного пошуку по документації.

6. Пошук у векторній базі: модуль RAG виконує пошук у векторній базі даних, сформованій на основі попередньо проіндексованої документації, та відбирає найбільш релевантні фрагменти.

7. Формування контексту: відібрані фрагменти документації передаються до мовної моделі (LLM) як контекст для побудови відповіді на запит користувача.

8. Генерація відповіді: LLM, використовуючи отриманий контекст із RAG-модуля, формує фінальну відповідь (Final answer), у якій узагальнює знайдену інформацію та, за потреби, може посилатися на конкретні фрагменти документації.

Розроблена архітектура забезпечує:

- коректне та масштабоване зберігання семантичних векторних представлень;
- швидкий пошук релевантного контексту;
- формування відповідей, заснованих на фактичних документах;
- можливість інтеграції у навчальні та корпоративні системи як універсального інтелектуального пошукового модуля.

Реалізація системи довела, що поєднання RAG-підходу та векторного пошуку забезпечує більш точні та обґрунтовані результати порівняно з традиційними методами пошуку.

Висновки

У роботі запропоновано архітектуру спеціалізованого ШІ-помічника, який поєднує методи семантичної індексації, векторні представлення текстових фрагментів та Retrieval-Augmented Generation. Запропоноване архітектурне рішення дало змогу поєднати класичні методи обробки текстових даних із сучасними мовними моделями, забезпечивши отримання точних та обґрунтованих відповідей на запитання користувачів у межах заданої предметної області.

Модульна структура системи передбачає чітке розділення відповідальності між компонентами: парсинг та нормалізація документації, формування векторних подань, векторний пошук, RAG-оркестрація та генерація фінальної відповіді. Такий підхід спростив розробку, налагодження та подальшу підтримку програмного коду, а також створив умови для незалежної еволюції окремих підсистем без порушення загальної логіки роботи.

Ключовим досягненням є побудова повноцінного конвеєра обробки документів: від завантаження та фрагментації тексту до збереження векторних представлень у базі даних і подальшого використання їх у процесі семантичного пошуку. Це дозволило перейти від поверхневого пошуку за ключовими словами до контекстно-орієнтованого вилучення знань, що безпосередньо підвищує якість відповідей ШІ-помічника.

Практична цінність розробленої системи полягає в можливості її інтеграції в існуючі інформаційні середовища організацій для роботи з внутрішньою документацією, регламентами, технічними описами чи навчальними матеріалами. Система може слугувати основою для побудови галузевих довідкових сервісів, навчальних платформ або внутрішніх консультантів для співробітників компаній.

Сформована архітектура має значний потенціал для подальшого розвитку. Завдяки модульному підходу та використанню векторного сховища знань система може бути розширена за рахунок підтримки нових предметних областей, підключення додаткових джерел даних, впровадження механізмів активного зворотного зв'язку від користувачів та адаптивного донавчання мовної моделі. Це забезпечує її актуальність і життєздатність у умовах швидкої еволюції технологій штучного інтелекту.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Lewis P. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks [Електронний ресурс] / P. Lewis, E. Perez, A. Piktus та ін. – 2020. – Режим доступу: <https://arxiv.org/abs/2005.11401>
2. FastAPI [Електронний ресурс] // Режим доступу: <https://fastapi.tiangolo.com/>
3. Vector similarity search for PostgreSQL [Електронний ресурс] // Режим доступу: <https://github.com/pgvector/pgvector>

Бурлаченко Артем Віталійович — студент групи 2ICT-24М, факультет інтелектуальних інформаційних

технологій та автоматизації, Вінницький національний технічний університет, Вінниця, e-mail: artemburlacenko95@gmail.com.

Жуков Сергій Олександрович — кандидат технічних наук, доцент кафедри системного аналізу та інформаційних технологій, Вінниця, e-mail: sazhukov@gmail.com.

Burlachenko Artem Vitaliiovych— student a student of group 2IST-24M, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia. E-mail: artemburlacenko95@gmail.com

Zhukov Serhii O. — Cand. Sc. (Eng.), Assistant Professor of the Department of Systems Analysis and Information Technologies, Vinnytsia, e-mail: sazhukov@gmail.com.