

DOI: <https://doi.org/10.36910/6775-2524-0560-2026-62-37>

УДК 347.77:007

Кабак Леонід Віталійович¹, к.т.н., доцент

<http://orcid.org/0000-0001-6267-1772>

Мороз Дмитро Максимович¹, Ph.D

<https://orcid.org/0000-0003-2577-3352>

Варех Нонна В'ячеславівна², к.н. соц. ком.

<https://orcid.org/0000-0002-2779-9225>

Сергієнко Олександр Романович¹, студент

¹Національний технічний університет «Дніпровська політехніка», м. Дніпро, Україна

²Університет «Метінвест політехніка», м. Запоріжжя, Україна

МЕТОДИ ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ СИСТЕМИ ЮРИДИЧНОЇ КОНСУЛЬТАЦІЇ

Кабак Л.В., Мороз Д.М., Варех Н.В., Сергієнко О.Р. Методи використання штучного інтелекту для системи юридичної консультації. Стаття присвячена розробці та вдосконаленню інтелектуальної інформаційної системи юридичних консультацій на основі архітектури Retrieval-Augmented Generation (RAG). У роботі детально проаналізовано ключові виклики сучасних інформаційно-правових систем, зокрема проблему «семантичного розриву» між побутовими запитам користувачів та формальною юридичною термінологією, а також недоліки традиційних методів інформаційного пошуку, заснованих на ключових словах. У статті обґрунтовано необхідність переходу до семантичних моделей пошуку, які забезпечують коректне відображення змісту правових норм у багатовимірному векторному просторі. У статті запропоновано гібридний підхід до пошуку юридичних документів, який поєднує векторний семантичний пошук, повнотекстове індексування та алгоритм ранжування Reciprocal Rank Fusion. Такий підхід дозволяє підвищити точність і стабільність видачі результатів завдяки компенсуванню недоліків окремих методів. Особливу увагу приділено механізму Smart Early Exit, що забезпечує миттєвий пошук у випадках, коли намір користувача вказує на конкретну статтю або норму закону. Науковою новизною роботи є розробка методу ієрархічного семантичного чанкінгу, який враховує структурні особливості українських нормативно-правових актів. Це дозволяє зберегти юридичний контекст під час векторизації та підвищити якість пошуку релевантних фрагментів. Запропонована мікросервісна архітектура системи забезпечує масштабованість, відмовостійкість та ефективний розподіл навантаження між модулями збору, обробки та генерації відповідей. Отримані результати демонструють, що поєднання гібридного пошуку, ієрархічного чанкінгу та нейромережевого реранкінгу дозволяє мінімізувати інформаційні втрати та підвищити достовірність юридичних консультацій. Перспективи подальших досліджень пов'язані з адаптацією системи до багатомовної обробки та інтеграції різних типів правової документації.

Ключові слова: штучний інтелект, архітектура RAG, семантичний чанкінг алгоритми ранжування, теорія множин, сегментація тексту, юридична консультація.

Kabak L., Moroz D., Varekh N., Serhiienko O. Methods of using artificial injector for a legal advice system. The article is devoted to the development and improvement of an intelligent legal advisory information system based on the Retrieval Augmented Generation (RAG) architecture. The paper provides a detailed analysis of the key challenges faced by modern legal information systems, in particular the problem of the "semantic gap" between users' everyday queries and formal legal terminology, as well as the shortcomings of traditional keyword-based information retrieval methods. The study substantiates the need to transition to semantic search models that ensure accurate representation of the content of legal norms in a multidimensional vector space. The article proposes a hybrid approach to legal document retrieval that combines vector-based semantic search, full-text indexing, and the Reciprocal Rank Fusion ranking algorithm. This approach improves the accuracy and stability of search results by compensating for the limitations of individual methods. Special attention is given to the Smart Early Exit mechanism, which enables instant retrieval in cases where the user's intent clearly indicates a specific article or legal provision. The scientific novelty of the work lies in the development of a hierarchical semantic chunking method that takes into account the structural features of Ukrainian regulatory legal acts. This makes it possible to preserve legal context during vectorization and improve the quality of retrieving relevant fragments. The proposed microservice-based system architecture ensures scalability, fault tolerance, and efficient load distribution among the modules responsible for data collection, processing, and response generation. The results demonstrate that combining hybrid search, hierarchical chunking, and neural network-based re-ranking minimizes information loss and enhances the reliability of legal consultations. Future research prospects are associated with adapting the system for multilingual processing and integrating various types of legal documentation.

Keywords: artificial intelligence, RAG architecture, semantic chunking, ranking algorithms, set theory, text segmentation, legal consultation.

Постановка наукової проблеми.

Стрімкий розвиток технологій штучного інтелекту (Artificial Intelligence, AI) та обробки природної мови (Natural Language Processing, NLP) докорінно змінює традиційні галузі, і сфера права не є винятком [1]. LegalTech (юридичні технології) трансформувалася з інструменту для оцифрування паперових архівів у потужну індустрію, що пропонує рішення для автоматизації складних юридичних процесів [2].

В умовах постійних змін у законодавстві України, особливо під час воєнного стану, громадяни та юристи стикаються з проблемою інформаційного перевантаження [3]. Традиційні пошукові системи, що базуються на ключових словах, часто не здатні знайти релевантну норму права, якщо запит сформульовано побутовою мовою. Тому розробка інтелектуальних систем класу RAG (Retrieval-Augmented Generation), здатних розуміти семантику запиту та надавати верифіковані відповіді з посиланнями на джерела, є вкрай актуальною науково-практичною задачею [4].

Аналіз досліджень.

На світовому рівні спостерігається перехід від використання простих баз даних до впровадження генеративних моделей (Generative AI). За даними аналітичних звітів (наприклад, Gartner та Stanford CodeX), ключовим трендом останніх років є використання великих мовних моделей (LLM) для створення «юридичних ко-пілотів» — асистентів, здатних не лише шукати інформацію, а й інтерпретувати її [5].

В Україні сфера LegalTech також демонструє значну динаміку, що зумовлено як високим рівнем цифровізації державних послуг (екосистема «Дія», відкриті реєстри), так і необхідністю швидкої адаптації до змін у законодавстві в умовах воєнного стану [6]. Український ринок представлений потужними інформаційно-правовими системами, такими як «ЛІГА:ЗАКОН», YouControl, Opendatabot, а також офіційним порталом Верховної Ради України (zakon.rada.gov.ua) [7].

Проте, аналіз сучасного стану виявив низку проблем:

1. Інформаційне перевантаження. Кількість нормативно-правових актів постійно зростає. Наприклад, офіційний портал ВР щоденно оновлюється десятками документів, що ускладнює їх ручний моніторинг.

2. Обмеженість ключового пошуку. Більшість існуючих національних систем використовують класичний повнотекстовий пошук (Full-Text Search). Це означає, що запит «покарання за хабар» може не знайти статтю, де фігурує лише термін «неправомірна вигода», оскільки відсутній семантичний зв'язок [8].

3. Високий поріг входження. Громадянам без юридичної освіти складно формулювати точні пошукові запити та інтерпретувати суху юридичну термінологію знайдених документів.

Отже, основною тенденцією розвитку LegalTech є створення інтелектуальних систем, здатних розуміти контекст запиту користувача природною мовою та надавати узагальнену відповідь з посиланнями на першоджерела. Саме на вирішення цієї задачі спрямована розробка систем класу RAG (Retrieval-Augmented Generation), які поєднують актуальність пошукових систем з аналітичними здібностями штучного інтелекту [9].

В інтелектуальних системах надання юридичних консультацій історично використовувалися різні підходи, кожен з яких має свої переваги та недоліки. Проведений аналіз дозволяє класифікувати їх наступним чином:

1. Експертні системи на основі правил (Rule-Based Systems). Це найстаріший підхід, що базується на деревах рішень («якщо А, то Б») [10]. Цей підхід має високу точність у вузьких сферах (наприклад, розрахунок податків або штрафів ПДР). Недоліками цього підходу є неможливість масштабування. Будь-яка зміна в законі вимагає переписування коду системи. Вони не розуміють вільного тексту користувача.

2. Пошукові системи на основі ключових слів (Keyword Search). Використовуються на порталах типу «Законодавство України» або в системах на кшталт Elasticsearch [11]. Цей підхід має швидкий пошук точних фраз, офіційність джерел. Недоліками цього підходу є відсутність семантичного розуміння. Користувач повинен знати точну назву закону або номер статті. Результатом є список документів, а не відповідь на питання.

3. Генеративні моделі загального призначення (General LLMs - ChatGPT, Claude, Gemini). Використання "чистих" нейромереж для юридичних питань. При використанні цього підходу є можливість вести діалог, пояснювати складні речі простою мовою [12]. Однак критична проблема «галюцинацій» (модель може вигадати неіснуючий закон). Знання моделей обмежені датою їх тренування (Knowledge Cutoff), тому вони не знають про закони, прийняті вчора чи сьогодні. Це неприпустимо в юриспруденції.

4. Системи RAG (Retrieval-Augmented Generation). Сучасний підхід, обраний як основа для даної роботи. Система спочатку знаходить релевантні фрагменти документів у базі знань (векторний пошук), а потім передає їх LLM для генерації відповіді [13]. При використанні цього підходу відповідь базується виключно на знайдених фактах, що мінімізує можливість галюцинації.

Мета статті. є підвищення ефективності та доступності правової допомоги шляхом розробки інтелектуальної інформаційної системи «ШІ-юрист» на основі архітектури RAG. Мінімізація надлишковості генерованої інформації разом з мінімізацією втрат, пов'язаних з процесом гібридного пошуку юридичних документів шляхом комбінації векторної схожості є ключовими аспектами запропонованого методу.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.

Однією з фундаментальних проблем, що обмежує ефективність традиційних інформаційно-пошукових систем (Information Retrieval, IR) у сфері права, є так званий "семантичний розрив". Цей феномен виникає через невідповідність між мовою запиту користувача (часто сформульованою побутовою лексикою) та спеціалізованою термінологією нормативно-правових актів.

У контексті українського законодавства ця проблема загострюється через багату морфологію мови та наявність синонімічних рядів, які не перетинаються лексично, але є тотожними юридично. Наприклад, користувач може шукати "покарання за побиття", тоді як Кримінальний кодекс України оперує термінами "тілесні ушкодження" різного ступеня тяжкості [14]. Класичні алгоритми BM25 або TF-IDF, що базуються на частоті спільних слів, у такому випадку привласняють документу нульову релевантність.

Дослідження показують, що в юридичному домені існують два типи невідповідностей:

1. Лексична невідповідність: Використання різних слів для опису одного поняття (наприклад, "аліменти" та "кошти на утримання").
2. Структурна невідповідність: Складність синтаксичних конструкцій закону, де одне речення може займати цілу сторінку і містити численні відсилання ("бланкетні норми") [15].

Застосування нейромережових моделей (Neural IR) дозволяє подолати цей бар'єр шляхом відображення слів у безперервний векторний простір, де семантично близькі поняття розташовуються поруч. Саме це обґрунтовує необхідність переходу від ключового пошуку до щільного векторного пошуку (Dense Retrieval) у сучасних LegalTech системах [16].

Для подолання проблеми актуальності даних та галюцинацій було розроблено архітектурний підхід RAG (Retrieval-Augmented Generation — Генерація з доповненим пошуком) [12]. Суть методу полягає у розділенні процесу генерації відповіді на два етапи: пошук релевантної інформації (Retrieval) та генерація відповіді на основі знайденого контексту (Generation).

В рамках даної роботи розглядається побудова RAG-системи, яка використовує **метод пошуку відповідних документів** який складається з наступних кроків:

1. Векторизація (Embedding): Тексти нормативно-правових актів розбиваються на логічні фрагменти (чанки) — наприклад, окремі статті або їх частини. Кожен чанк перетворюється на числовий вектор (embedding) фіксованої розмірності за допомогою моделей-енкодерів. Вектор відображає семантичний зміст тексту: фрагменти зі схожим змістом будуть знаходитись поруч у багатовимірному просторі.
2. Векторне сховище (Vector Database): Згенеровані вектори зберігаються у спеціалізованій базі даних (у даному проекті використовується PostgreSQL з розширенням pgvector). Це дозволяє виконувати пошук найближчих сусідів (Nearest Neighbor Search) за метрикою косинусної подібності (Cosine Similarity).
3. Гібридний пошук (Hybrid Search): Для підвищення точності юридичного пошуку недостатньо лише векторного співставлення, оскільки воно може пропустити точні вхідження номерів статей. Тому доцільно використовувати гібридний підхід, що поєднує:
Семантичний пошук (Vector Search): Знаходить документи за змістом (наприклад, запит «вбивство» знайде статтю 115 ККУ, навіть якщо слово «вбивство» в запиті не вжито, а описано ситуацію).
Повнотекстовий пошук (Full-Text Search / SQL Lookup): Забезпечує точне знаходження за ключовими словами або номерами статей (наприклад, «стаття 307»).
4. Реранкінг (Reranking): Етап, на якому відібрані кандидатури документів переоцінюються більш потужною моделлю (Cross-Encoder) для визначення найбільш релевантних фрагментів [18], які будуть передані в контекст LLM.
5. Генерація (Generation): На фінальному етапі сформований контекст (знайдені фрагменти законів) разом із запитом користувача та системним промптом (інструкцією) передається в LLM. Модель виступає в ролі аналітика, який не «згадує» знання, а аналізує надані їй документи.

Такий підхід забезпечує прозорість (Explainability) — система може надати точні посилання на джерела, використані для відповіді, що є критичною вимогою для юридичних систем.

На базі розробленого методу було спроектовано інформаційну систему, було проаналізовано вимоги до масштабованості, відмовостійкості та простоти підтримки програмного забезпечення. Враховуючи суттєві відмінності у вимогах до обчислювальних ресурсів для різних підзадач (I/O-інтенсивний парсинг проти CPU-інтенсивної векторизації), було обрано мікросервісний архітектурний патерн. Такий підхід забезпечує ізоляцію процесів та дозволяє масштабувати окремі компоненти системи незалежно один від одного.

Архітектуру системи було спроектовано та декомпозовано на п'ять ключових незалежних модулів, які взаємодіють через спільну базу даних PostgreSQL. Структурна схема взаємодії компонентів системи наведена на рис. 1.

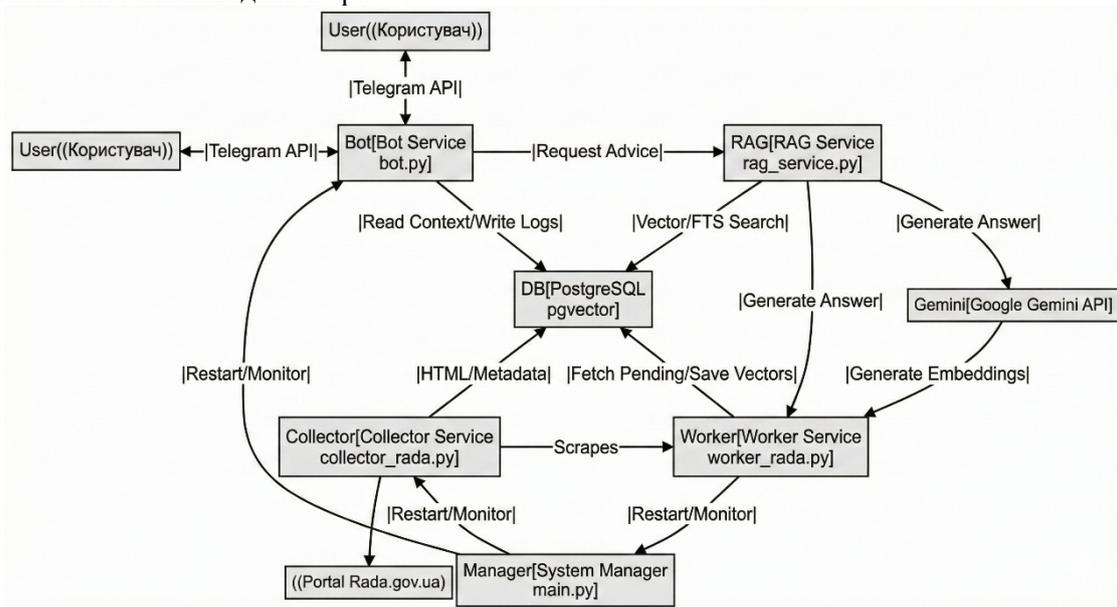


Рис. 1. Структурна схема мікросервісної архітектури системи

Запропонована архітектура складається з наступних компонентів:

1. Модуль оркестрації (System Manager / main.py): Виконує функцію системного наглядача (Watchdog). Він відповідає за запуск, моніторинг життєвого циклу інших процесів та їх автоматичний перезапуск у разі аварійного завершення. Це критично важливо для стабільності, оскільки робота з мережею та веб-драйверами (Selenium/Playwright) є потенційно нестабільною і може призводити до зависання процесів.
2. Модуль збору даних (Collector / collector_rada.py): Відповідає за взаємодію із зовнішнім джерелом (портал ВРУ), моніторинг нових документів та формування черги завдань. Модуль працює в автономному режимі, періодично опитуючи джерело даних.
3. Модуль обробки (Worker / worker_rada.py): Виконує ресурсоємні операції: очищення HTML, розбивку тексту на фрагменти (chunking) та генерацію векторних представлень (embeddings) за допомогою Google Gemini API. Винесення цієї логіки в окремий процес дозволяє уникнути блокування інтерфейсу користувача під час обробки великих масивів даних.
4. Інтерфейсний модуль (Bot / bot.py): Забезпечує асинхронну взаємодію з користувачем через Telegram API, керування станами діалогу (FSM) та передачу запитів до сервісу пошуку.
5. Сервіс RAG (Retrieval-Augmented Generation / rag_service.py): Реалізує логіку гібридного пошуку (векторний + повнотекстовий), рананкінгу результатів та генерації відповіді.

Метод ієрархічного чанкінгу (Hierarchical Chunking) юридичних текстів

Традиційні методи розбиття тексту на фрагменти (chunking) з фіксованим розміром вікна (fixed-size window) виявилися неефективними для нормативно-правових актів. Розрив тексту посеред речення або відокремлення санкції статті від її диспозиції призводить до втрати контексту, що робить векторизацію «шумною». У модулі worker_rada.py розроблено алгоритм ієрархічного семантичного чанкінгу (Hierarchical Semantic Chunking), адаптований під структуру українського законодавства. Модель процесу трансформації сирого тексту у векторизовані чанки наведена на рис. 2.

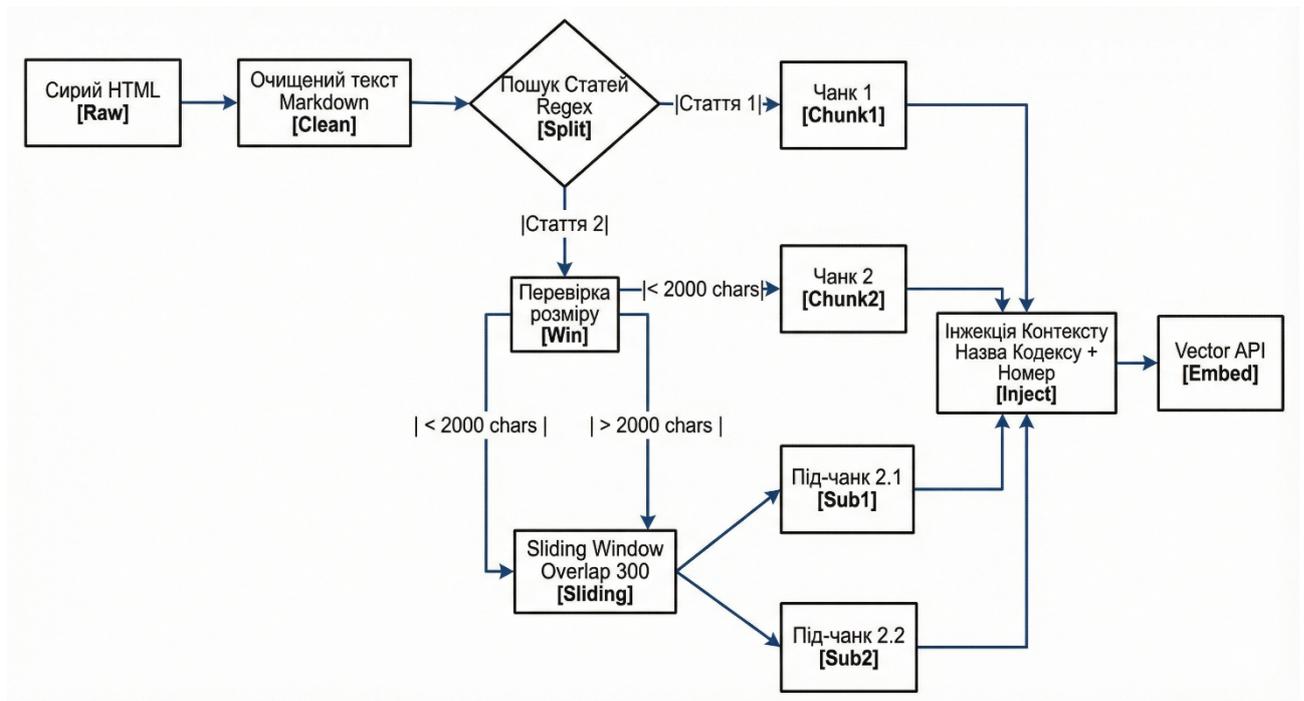


Рис. 2 Метод ієрархічного чанкінгу

Однією з головних проблем класичних RAG-систем у юридичному домені є обмеженість "чистого" векторного пошуку, який може ігнорувати точні ключові слова. Для вирішення цієї проблеми реалізовано багаторівневий конвеєр (Pipeline) гібридного пошуку. Векторний пошук базується на обчисленні косинусної подібності (Cosine Similarity) між вектором запиту Q та вектором документа D :

$$sim(Q, D) = \frac{\sum_{i=1}^n Q_i D_i}{\sqrt{\sum_{i=1}^n Q_i^2} \sqrt{\sum_{i=1}^n D_i^2}} \quad (1)$$

Процес пошуку реалізовано у вигляді конвеєра, зображеного на рис. 3.

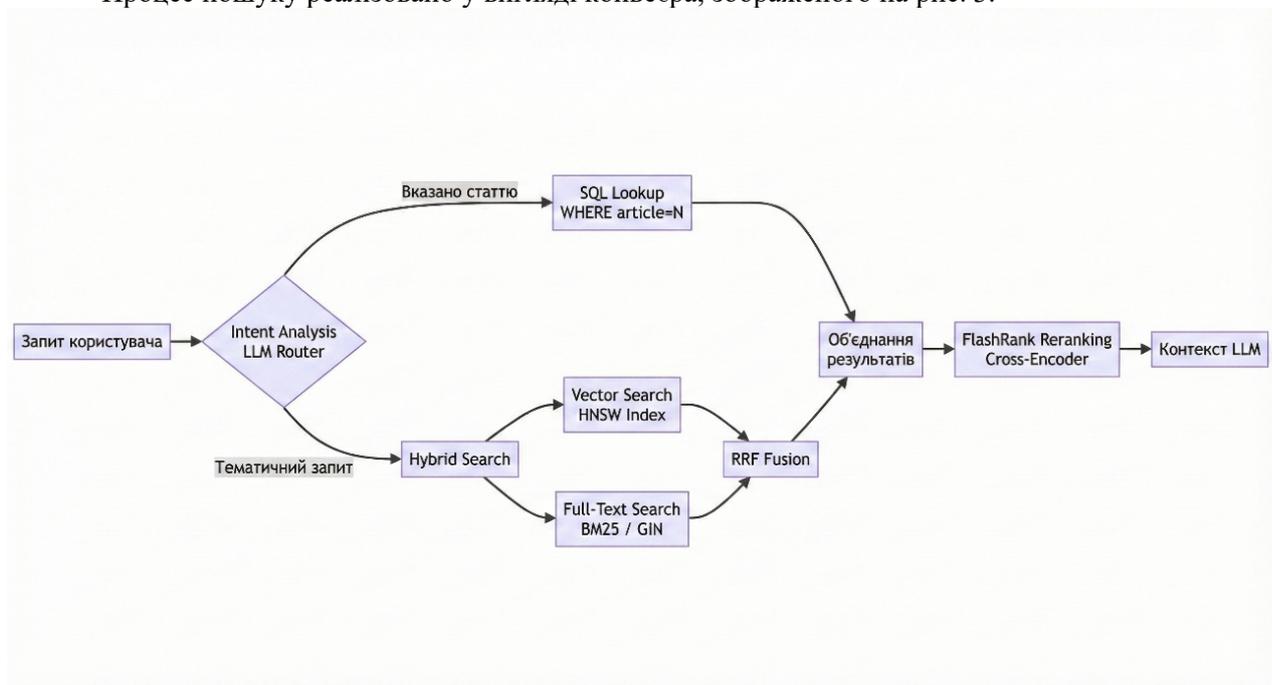


Рис. 3 Конвеєр (Pipeline) гібридного пошуку та генерації відповіді

Для визначення семантичної близькості між запитом користувача та фрагментами законодавства використовується метрика косинусної подібності (Cosine Similarity) [18]. Вектори

генеруються моделлю text-embedding-004 з розмірністю $n = 768$. Відстань між вектором запиту A та вектором документа B у багатовимірному просторі обчислюється як:

$$\text{similarity}(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (2)$$

У програмній реалізації (файл rag_service.py) використовується оператор \Leftrightarrow розширення `pgvector`, який обчислює косинусну відстань ($1 - \text{similarity}$).

Реалізація гібридного пошуку в модулі rag_service.py базується на алгоритмі злиття рангів RRF (Reciprocal Rank Fusion) [6]. Цей метод дозволяє об'єднати результати двох принципово різних пошукових механізмів без необхідності складної калібровки їхніх оцінок.

Злиття результатів виконується за формулою:

$$\text{Score}_{RRF}(d) = \sum_{m \in M} \frac{1}{k + \text{rank}_m(d)} \cdot w_m \quad (3)$$

3)

де:

- M — множина методів пошуку (Vector, FTS);
- $\text{rank}_m(d)$ — позиція документу у видачі методу $\$m\$$;
- k — константа згладжування (в системі $k=60$);
- w_m — вага методу (емпірично встановлено $w_{fts} = 1.2$, $w_{vec} = 1.0$ для пріоритету текстових збігів).

Процес пошуку включає етап "Smart Early Exit": якщо за допомогою аналізу намірів (`analyze_query_intent`) виявлено запит на конкретну статтю (наприклад, "ст. 307 ККУ"), виконується точний SQL-запит, і система негайно повертає результат, минаючи етап векторного пошуку.

Для об'єднання результатів векторного та лексичного пошуку застосовано метод Reciprocal Rank Fusion. На відміну від методів, що базуються на лінійній комбінації нормалізованих оцінок (score-based fusion), RRF оперує виключно рангами документів, що робить його стійким до різного масштабу оцінок (наприклад, косинусна відстань варіюється від 0 до 1, тоді як оцінка BM25 може бути > 100).

Математична модель RRF для документа $d \in D$ визначається як сума обернених рангів:

$$\text{RRFcore}(d) = \sum_{r \in R} \frac{1}{k + r(d)} \quad (4)$$

де:

- R — множина ранжувальників (у нашому випадку: R_1 — семантичний пошук, R_2 — повнотекстовий пошук);
- $r(d)$ — позиція (ранг) документа d у видачі конкретного ранжувальника;
- k — константа згладжування.

У розробленій системі обрано значення $k = 60$, що є емпіричним стандартом індустрії. Це значення запобігає домінуванню документів, які мають надзвичайно високий рейтинг в одній системі, але відсутні в іншій. Таким чином, документ, який знаходиться на 10-му місці в обох списках, отримає вищий фінальний бал, ніж документ, який є 1-м в одному списку і відсутній в іншому. Це критично для юридичного пошуку, де важливо знайти документи, які є релевантними як за змістом, так і за точною термінологією.

Гібридний пошук ефективно відбирає кандидатів, але не завжди ідеально їх сортує. Для вирішення цієї проблеми у фінальній стадії конвеєра (rag_service.py) інтегровано модуль Reranking на базі бібліотеки FlashRank.

Використовується модель Cross-Encoder (ms-marco-MultiBERT-L-12). На відміну від Bi-Encoder, який кодує запит і документ окремо, Cross-Encoder приймає на вхід пару "Запит + Документ" і оцінює їхню взаємодію на рівні механізму уваги (Self-Attention). Це дозволяє моделі переоцінити релевантність документів (confidence score від 0 до 1) і підняти нагору ті, що містять пряму відповідь на запитання користувача.

Висновки та перспективи подальшого дослідження

У результаті проведеної роботи було удосконалено метод гібридного пошуку юридичних документів шляхом комбінації векторної схожості та повнотекстового ранжування з використанням алгоритму "розумного раннього виходу" (Smart Early Exit), що дозволило підвищити релевантність видачі вірних юридичних консультацій користувачам. Тако ж набув подальшого розвитку метод

семантичного чанкінгу (сегментації) текстів, який, на відміну від існуючих, враховує ієрархічну структуру українських нормативно-правових актів (розділ, стаття, частина), що забезпечує збереження юридичного контексту.

Таким чином, результати роботи підтверджують гіпотезу про те, що комбінація гібридного пошуку, ієрархічного чанкінгу та нейромережевого реранкінгу на базі архітектури RAG дозволяє створити надійну та точну систему правової допомоги, мінімізуючи ризики, властиві сучасним мовним моделям.

Перспективами подальшого дослідження є розширення архітектури RAG для багатомовної підтримки та адаптація системи для роботи з багатомовними нормативно-правовими актами (українська, англійська, інші мови), що актуально для міжнародних компаній та юристів.

Список бібліографічного опису

1. Russell S. J., Norvig P. Artificial Intelligence: A Modern Approach: Pearson, 2020. 1132 p. URL: <https://people.engr.tamu.edu/guni/csce625/slides/AI.pdf> (дата звернення: 10.12.2025).
2. Susskind R. Tomorrow's Lawyers: An Introduction to your Future (Third Edition): Oxford University Press, 2023. 240 p. <https://doi.org/10.1093/9780192864727.001.0001>
3. Законодавство України в умовах воєнного стану: інформаційно-аналітичний огляд Верховна Рада України. URL: <https://zakon.rada.gov.ua/> (дата звернення: 10.12.2025).
4. Lewis P., Perez E., Aleksandra Piktus Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks: Cornell University: <https://doi.org/10.48550/arXiv.2005.11401>
5. Top-10 стратегічних технологічних трендів Gartner на 2024 рік Gartner, 2023. URL: <https://www.gartner.com/en/articles/gartner-top-10-strategic-technology-trends-for-2024> (дата звернення: 12.12.2025).
6. Цифрова держава: як працює екосистема Дія URL: <https://diia.gov.ua/> (дата звернення: 12.12.2025).
7. Про затвердження Положення про веб-портал Верховної Ради України : Розпорядження Голови Верховної Ради України від 19.05.2011 р. № 388. URL: <https://zakon.rada.gov.ua/laws/show/462/01-%D1%80%D0%B3#Text> (дата звернення: 12.12.2025).
8. Robertson I. S., Zaragoza H. The Probabilistic Relevance Framework: BM25 and Beyond: Foundations and Trends in Information Retrieval Vol. 3, No. 4 (2009) 333–389 p. <https://doi.org/10.1561/1500000019>
9. Gao Y., Xion Y., Xigun G. ... Retrieval-Augmented Generation for Large Language Models: A Survey: arXiv preprint arXiv:2312.10997, 2023. <https://doi.org/10.48550/arXiv.2312.10997>
10. Poppel J. A Pragmatic Legal Expert System: Australian National University: 1996. 375 p. URL: <https://iuristebi.wordpress.com/wp-content/uploads/2011/07/a-pragmatic-legal-expert-system.pdf> (дата звернення: 12.12.2025).
11. Gormley C., Tong Z. Elasticsearch: A Distributed Real-Time Search and Analytics Engine, The Definitive Guide: 2015. 686 p. URL: https://hlszny.com/booksAndPapers/buckets/b8_IT/elasticsearch-the-definitive-guide.pdf (дата звернення: 12.12.2025).
12. Achiam J, Adler S., Agarwal S. ... GPT-4 Technical Report/ OpenAI. — arXiv preprint arXiv:2303.08774, 2023. <https://doi.org/10.48550/arXiv.2303.08774>
13. Shuster K., Spenser P. Retrieval Augmentation Reduces Hallucination in Conversation: arXiv:2104.07567: 2021. <https://doi.org/10.48550/arXiv.2104.07567>
14. Кримінальний кодекс України : Закон України від 05.04.2001 р. № 2341-III // Відомості Верховної Ради України.: 2001.: № 25-26. С. 131. URL: <https://zakon.rada.gov.ua/laws/show/2341-14#Text> (дата звернення: 12.12.2025).
15. Ashley K. D. Artificial Intelligence and Legal Analytics: New Tools for Law Practice in the Digital Age: Cambridge University Press, 2017. 426 p.
16. Karpukhin V., Oguz B., Min S., Lewis P., Wu L., Edunov S., Chen D., and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, P. 6769–6781. <https://doi.org/10.18653/v1/2020.emnlp-main.550>
17. Nogueira R., Cho K. Passage Re-ranking with BERT.: arXiv preprint arXiv:1901.04085, 2019. <https://doi.org/10.48550/arXiv.1901.04085>
18. Сінгхал А. Modern Information Retrieval: A Brief Overview: Bulletin of the IEEE Computer Society Technical Committee on Data Engineering — 2001. — Т. 24, № 4. — P. 35–43. URL: <https://www.cs.columbia.edu/~gravano/Qual/Papers/singhal.pdf> (дата звернення: 12.12.2025).

References

1. Russell S. J., Norvig P. Artificial Intelligence: A Modern Approach: Pearson, 2020. 1132 p. URL: <https://people.engr.tamu.edu/guni/csce625/slides/AI.pdf> (дата звернення: 10.12.2025).
2. Susskind R. Tomorrow's Lawyers: An Introduction to your Future (Third Edition): Oxford University Press, 2023. 240 p. <https://doi.org/10.1093/9780192864727.001.0001>
3. Законодавство України в умовах воєнного стану: інформаційно-аналітичний огляд Верховна Рада України. URL: <https://zakon.rada.gov.ua/> (дата звернення: 10.12.2025).
4. Lewis P., Perez E., Aleksandra Piktus Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks: Cornell University: <https://doi.org/10.48550/arXiv.2005.11401>
5. Top-10 stratehichnykh tekhnolohichnykh trendiv Gartner na 2024 rik Gartner, 2023. URL: <https://www.gartner.com/en/articles/gartner-top-10-strategic-technology-trends-for-2024> (дата звернення: 12.12.2025).
6. Tsyfrova derzhava: yak pratsiuie ekosystema Diia URL: <https://diia.gov.ua/> (дата звернення: 12.12.2025).

7. Pro zatverdzhennia Polozhennia pro veb-portal Verkhovnoi Rady Ukrainy : Rozporiadzhennia Holovy Verkhovnoi Rady Ukrainy vid 19.05.2011 r. № 388. URL: <https://zakon.rada.gov.ua/laws/show/462/01-%D1%80%D0%B3#Text> (data zvernennia: 12.12.2025).
8. Robertson I S., Zaragoza H. The Probabilistic Relevance Framework: BM25 and Beyond: Foundations and TrendsR in Information Retrieval Vol. 3, No. 4 (2009) 333–389 p. <https://doi.org/10.1561/1500000019>
9. Gao Y., Xion Y., Xigun G. ... Retrieval-Augmented Generation for Large Language Models: A Survey: arXiv preprint arXiv:2312.10997, 2023. <https://doi.org/10.48550/arXiv.2312.10997>
10. Popple J. A Pragmatic Legal Expert System: Australian National University: 1996. 375 p. URL: <https://iuristebi.wordpress.com/wp-content/uploads/2011/07/a-pragmatic-legal-expert-system.pdf> (data zvernennia: 12.12.2025).
11. Gormley C., Tong Z. Elasticsearch: A Distributed Real-Time Search and Analytics Engine, The Definitive Guide: 2015. 686 p. URL: https://hlaszny.com/booksAndPapers/buckets/b8_IT/elasticsearch-the-definitive-guide.pdf (data zvernennia: 12.12.2025).
12. Achiam J, Adler S., Agarwal S. ... GPT-4 Technical Report/ OpenAI. — arXiv preprint arXiv:2303.08774, 2023. <https://doi.org/10.48550/arXiv.2303.08774>
13. Shuster K., Spenser P. Retrieval Augmentation Reduces Hallucination in Conversation: arXiv:2104.07567: 2021. <https://doi.org/10.48550/arXiv.2104.07567>
14. Kryminalnyi kodeks Ukrainy : Zakon Ukrainy vid 05.04.2001 r. № 2341-III // Vidomosti Verkhovnoi Rady Ukrainy.: 2001.: № 25-26. S. 131. URL: <https://zakon.rada.gov.ua/laws/show/2341-14#Text> (data zvernennia: 12.12.2025).
15. Ashley K. D. Artificial Intelligence and Legal Analytics: New Tools for Law Practice in the Digital Age: Cambridge University Press, 2017. 426 p.
16. Karpukhin V., Oguz B., Min S., Lewis P., Wu L., Edunov S., Chen D., and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), P. 6769–6781. <https://doi.org/10.18653/v1/2020.emnlp-main.550>
17. Nogueira R., Cho K. Passage Re-ranking with BERT.: arXiv preprint arXiv:1901.04085, 2019. <https://doi.org/10.48550/arXiv.1901.04085>
18. Sinhkhal A. Modern Information Retrieval: A Brief Overview: Bulletin of the IEEE Computer Society Technical Committee on Data Engineering — 2001. — T. 24, № 4. — P. 35–43. URL: <https://www.cs.columbia.edu/~gravano/Qual/Papers/singhal.pdf> (data zvernennia: 12.12.2025).

Історія статті:

Отримано: 11.02.2026 Доопрацьовано: 09.03.2026 Прийнято до друку: 23.03.2026 Опубліковано: 29.03.2026