ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ БІЗНЕС-КОЛЕДЖ

Системи Штучного Інтелекту на основі технології RAG

Д.т.н., професор Сергій Заболотній
zabolotnii.serhii@csbc.edu.ua

** засновано на матеріалах статті "A Taxonomy of Retrieval Augmented Generation" by Abhinav Kimothi (Medium)*

Вступ до Штучного Інтелекту (ШІ) та RAG

- **Штучний інтелект (ШІ)** — галузь комп'ютерних наук, що створює інтелектуальні системи, здатні виконувати складні завдання, такі як розпізнавання мовлення, аналіз текстів, розпізнавання образів, та ухвалення рішень. ШІ вже застосовується у різних сферах: медицині, бізнесі, освіті, автономному транспорті, фінансах та ін.

- Особливе місце займає **Generative AI** — технології, що дозволяють автоматично створювати тексти, зображення, відео та інший контент на основі великих обсягів навчальних даних.

- **Retrieval Augmented Generation (RAG)** — одна з новітніх технологій у сфері генеративного ШІ, яка поєднує потужність великих мовних моделей (LLM) із доступом до зовнішніх баз знань для отримання більш точних і актуальних відповідей.

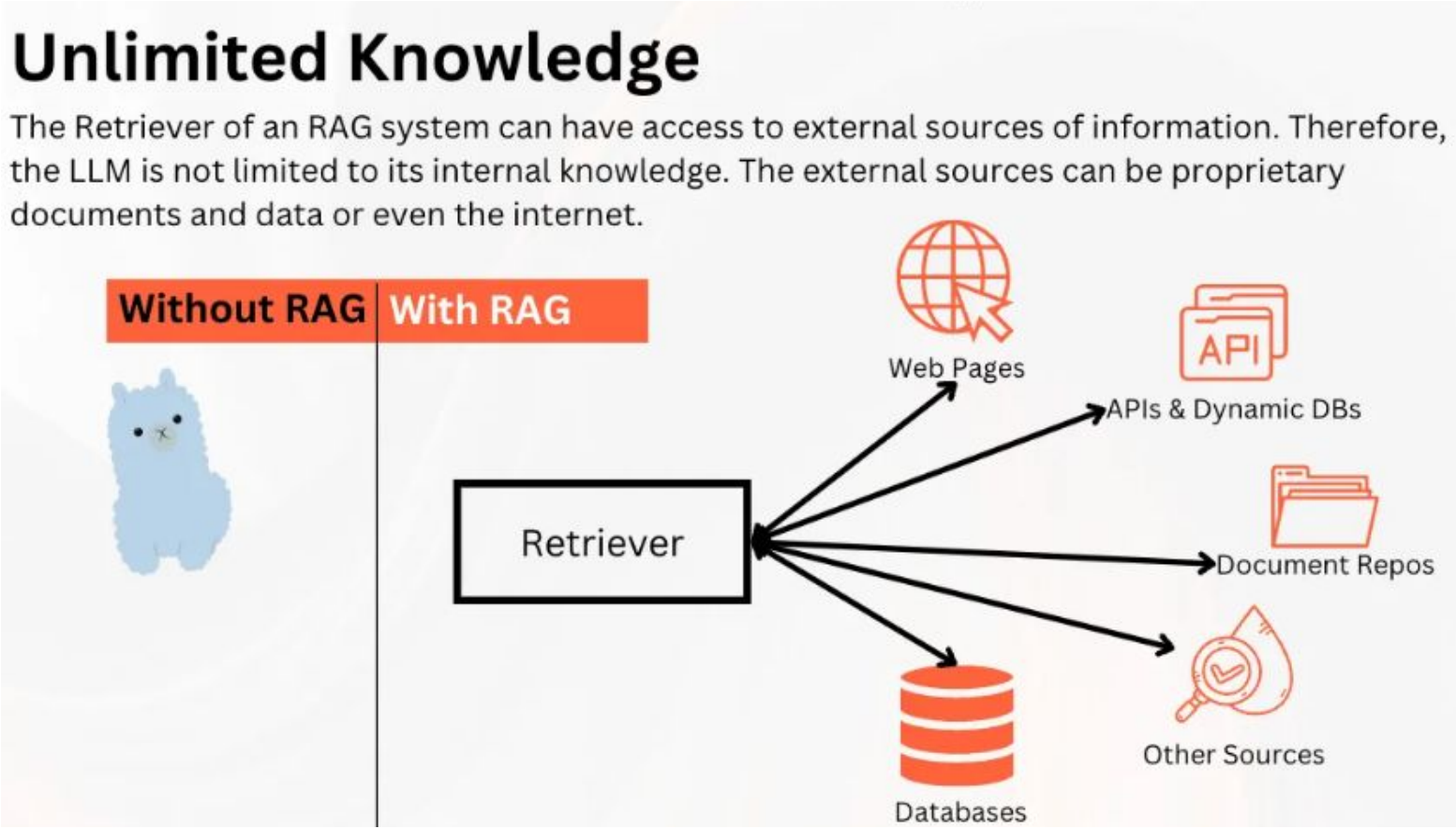
Обмеження сучасних мовних моделей (LLMs)

- **Застарілість даних**
  - Навчальні набори не оновлюються в реальному часі
  - Відсутність актуальних даних після "дата-відсічки" (Knowledge Cut-off)
- **Обмежений контекст**
  - Моделі мають фіксований розмір контексту (контекстне вікно)
  - Обмеження на довжину вхідного тексту
- **Галюцинації (hallucinations)**
  - Генерація неправдивих, але переконливих відповідей
  - Відсутність механізму перевірки достовірності фактів
- **Відсутність доступу до приватних даних**
  - Навчання тільки на публічних джерелах
  - Неможливість відповіді на питання з корпоративної чи конфіденційної інформації

Обмеженість контекстного вікна LLM

Model Name	Developer	Knowledge Cut-off Date	Context Window (Tokens)
Claude 3	Anthropic	March 2024	200k tokens
GPT-4o	OpenAI	April 2023	128k tokens
LLaMA 3.1	Meta	June 2024	128k tokens
PaLM 2	Google DeepMind	April 2023	32k tokens
Gemini 1.5 Pro	Google DeepMind	Early 2024	256k tokens
Claude 2	Anthropic	Early 2023	100k tokens
Mistral	Mistral AI	2023	8k tokens
Falcon 40B	TII	March 2023	2k tokens
BLOOM	BigScience	Early 2022	2k tokens
GPT-NeoX-20B	EleutherAI	April 2022	2k tokens

Чим може допомогти RAG?



Концепція Retrieval Augmented Generation (RAG)

Retrieval (Пошук):

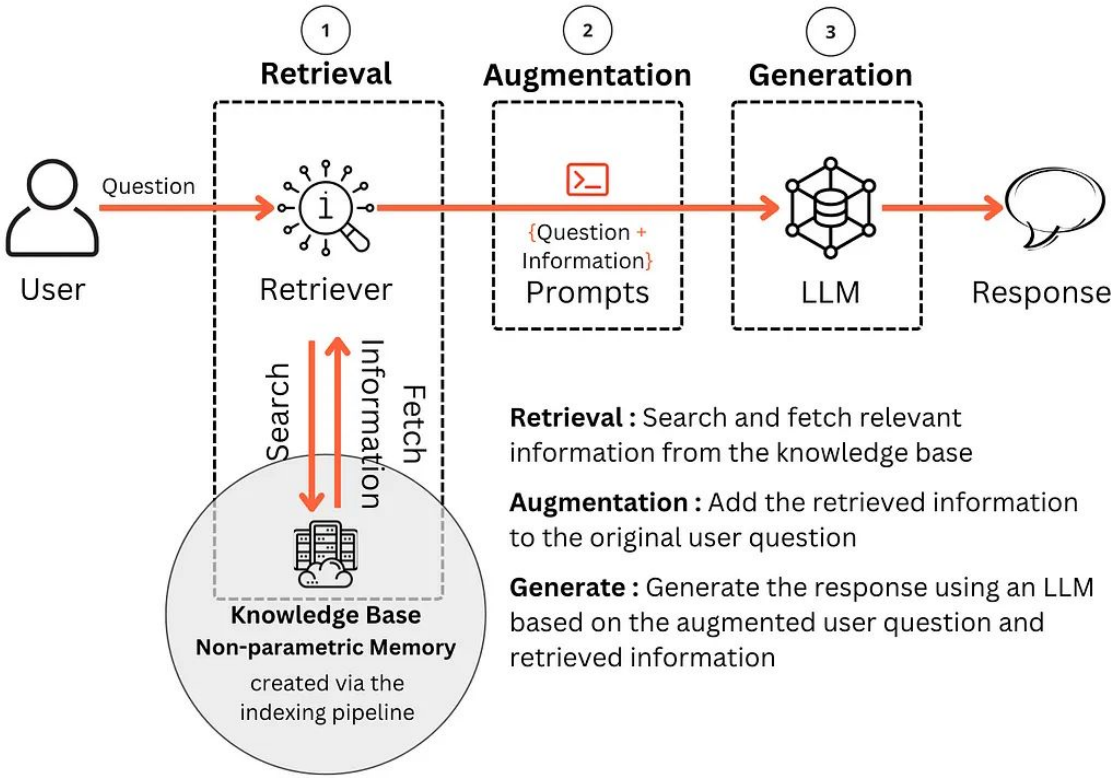
- Пошук релевантної інформації у зовнішній базі знань за запитом користувача.

Augmentation (Доповнення):

- Доповнення вихідного запиту користувача знайденою інформаціє для створення повнішого контексту.

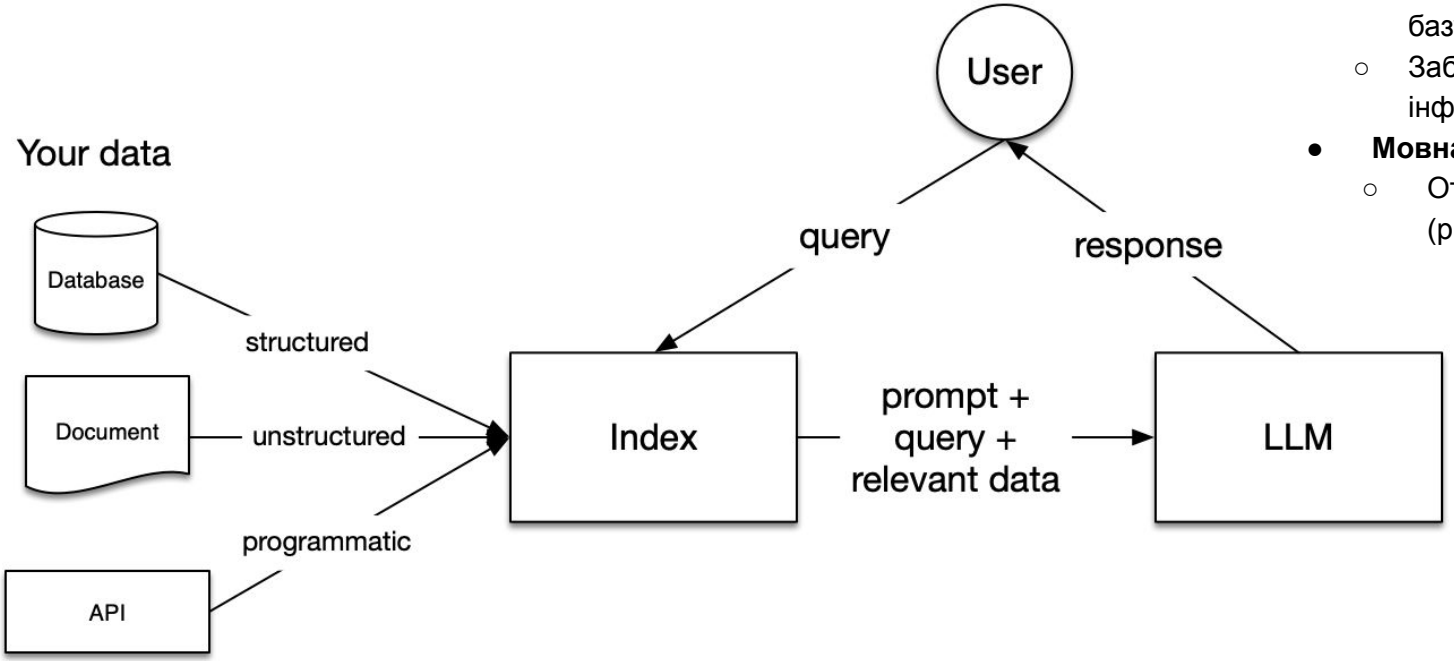
Generation (Генерація):

- Генерація фінальної відповіді мовно моделлю (LLM), що спирається на доповнений контекст



Структура та компоненти RAG-систем

- **Джерела даних**
  - Структуровані (бази даних)
  - Неструктуровані (документи, тексти)
  - Програмні (API)
- **Індекс (Index)**
  - Об'єднує різнорідні дані у єдину базу знань
  - Забезпечує ефективний пошук інформації
- **Мовна модель (LLM)**
  - Отримує доповнений запит (prompt + дані)

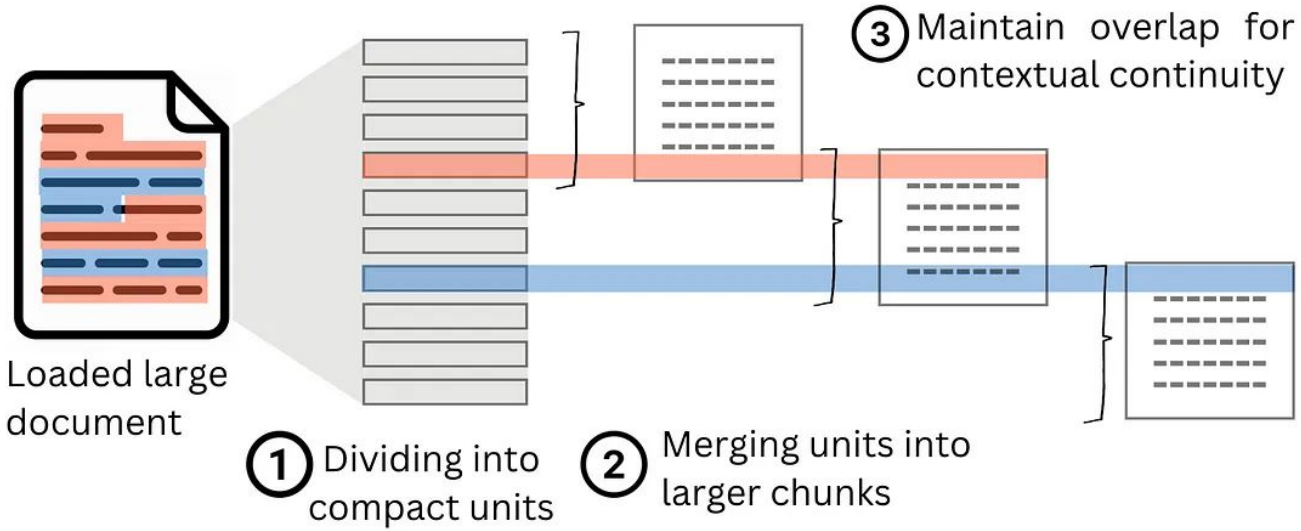


Формування бази знань (Knowledge Base)

- **Збір інформації:** вибір джерел для бази знань – корпоративні документи, довідники, статті, веб-дані, вікі тощо. Дані мають бути релевантними тематиці запитів.
- **Попередня обробка:** очистка текстів від зайвого, структуризація; додавання метаданих (наприклад, дата, автор, категорія) для покращення подальшого пошуку та фільтрації.
- **Розбиття на фрагменти:** великі тексти діляться на малі логічні шматки (chunks), щоб модель могла ефективно оперувати ними. Наприклад, розбивка по абзацах або за фіксованим розміром (скажімо, 200-500 слів).
- **Векторизація даних:** кожен фрагмент перетворюється на числове векторне представлення (embedding) за допомогою модельного перетворювача. Ці embeddings зберігаються у векторній базі даних для подальшого пошуку.
- **Оновлення знань:** база знань може періодично доповнюватися новими даними або оновлюватися, щоб система залишалася актуальною.

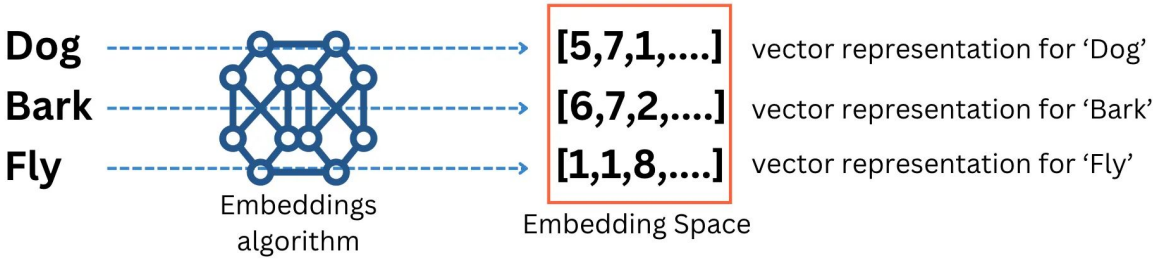
Методи індексації знань

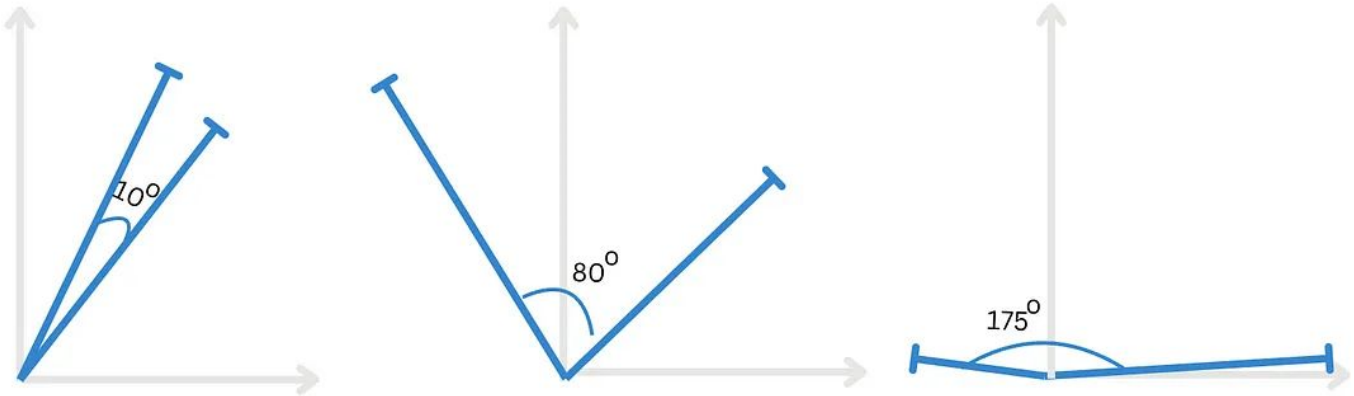
- **Chunking (розбиття на фрагменти):** стратегія індексації, коли документи діляться на менші фрагменти. Фрагменти можуть формуватися за фіксованим розміром (наприклад, N символів/слів) або за семантичними межами (розділи, абзаци). Можливе часткове перекриття між фрагментами (sliding window) для збереження контексту.



- **Embeddings (вбудовування):** для кожного текстового фрагмента обчислюється embedding – багатовимірний вектор, що відображає семантичний зміст. Подібні за змістом тексти матимуть близькі вектори. Приклад моделей для embedding: Sentence-BERT, OpenAI Ada2.
- **Векторна база даних:** спеціалізована БД для зберігання embeddings (наприклад, Chroma, FAISS, Pinecone). Забезпечує швидкий пошук найбільш схожих векторів (і відповідних їм фрагментів тексту) за заданим вектором запиту.

Векторне представлення та косинусна подібність





Cos 10 =0.985
Close to 1
Very Similar

Cos 80 =0.173
Close to 0
Unrelated

Cos 175 =-0.996
Close to -1
Opposite

Алгоритми пошуку

- **TF-IDF:** традиційний підхід до пошуку, що оцінює важливість слів у документі. Враховує частоту терміну (TF) у документі та зворотну частоту в корпусі (IDF). Використовується для базового ранжування документів за схожістю з запитом.

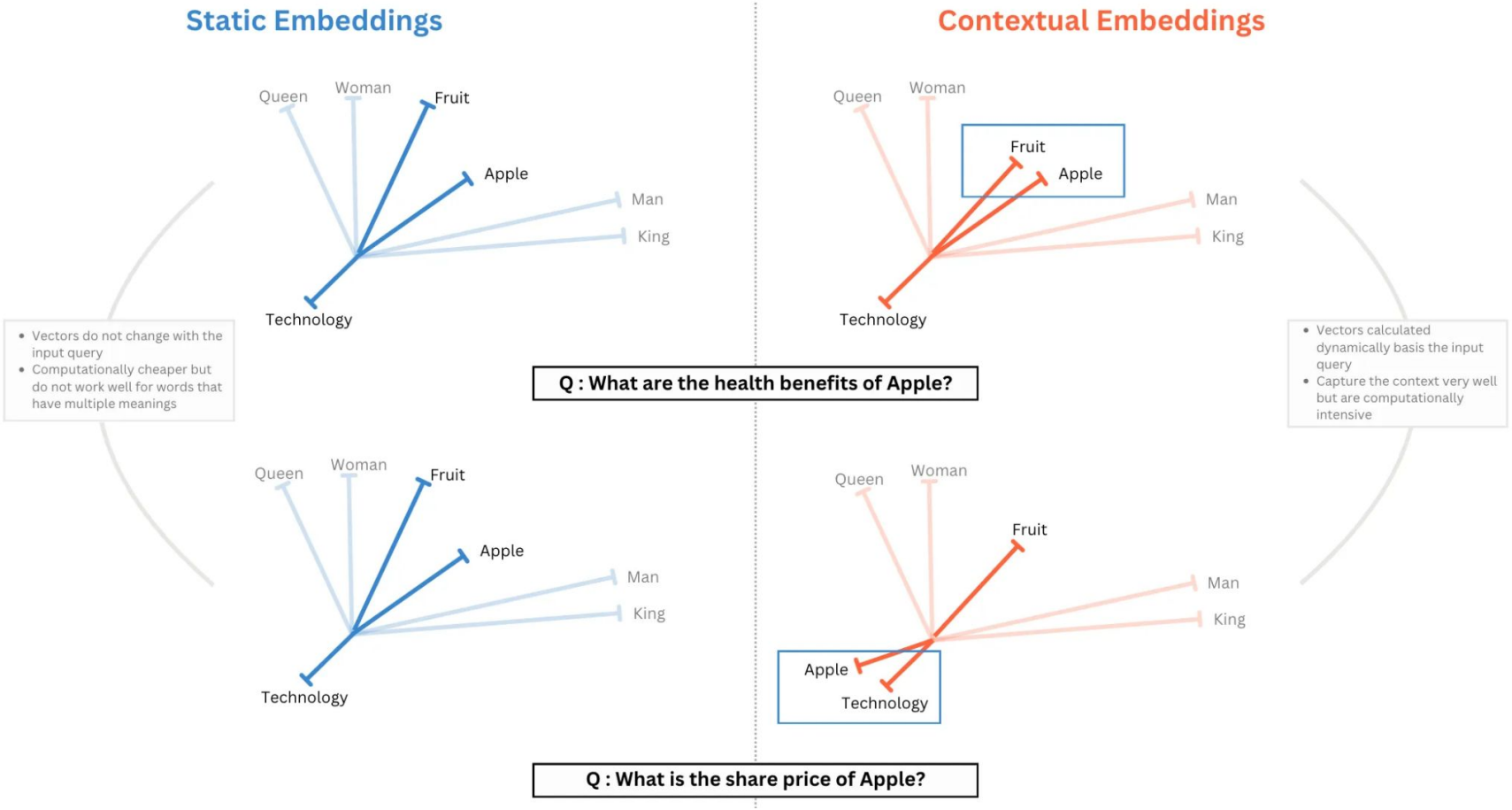
- **BM25:** покращений алгоритм пошуку на основі TF-IDF. Враховує довжину документа і частоту термінів, надаючи більш збалансовану оцінку релевантності. Стандарт для класичного лексичного пошуку в інформаційних системах.

- **Sparse vs Dense Retrieval:** *Sparse (розріджений) пошук* – методи на кшталт TF-IDF/BM25, де документи представлені як великі розріджені вектори за словами. *Dense (щільний) пошук* – методи на основі embeddings, де і запит, і документи представлені компактними семантичними векторами.

- **Dense Retrieval:** семантичний пошук, що використовує порівняння embeddings. Дозволяє знаходити релевантні тексти за змістом, навіть якщо ті не містять буквальних збігів зі словами запиту. Вимірюється *косинусна схожість* або інша метрика відстані між векторами запиту і документа.

- **Hybrid Retrieval:** гібридний підхід, що поєднує результати sparse- та dense-пошуку. Наприклад, спочатку відбір кандидатів BM25, а потім rerank за допомогою embeddings; або об’єднання списків результатів двох підходів. Це дає змогу врахувати і точні збіги, і семантичну близькість для кращої релевантності.

Статичні та контекстні вбудовування



Генерація відповідей (роль LLM)

- **Використання LLM:** на етапі генерації велика мовна модель отримує запит користувача разом зі знайденими фрагментами знань (контекстом) та генерує підсумкову відповідь. Модель комбінує власні знання та наданий контекст, формуючи зв’язний текст.

- **Prompt Engineering:** конструювання вхідного промпту для LLM, щоб спрямувати її роботу. До промпту включають: саме питання користувача, релевантні фрагменти з бази знань, а також інструкції (наприклад, «надай відповідь, спираючись лише на наведений текст»). Від якісно сформульованого промпту залежить коректність і повнота відповіді.

- **Contextual Prompting:** підказки з урахуванням контексту – модель отримує не тільки питання, а й відповідний контекст. Це знижує ймовірність галюцинацій, оскільки LLM опирається на факти з наданих джерел. Важливо забезпечити, щоб у контекст потрапила максимально корисна інформація (тому потрібен надійний пошук).

- **Післягенераційні перевірки:** можна налаштувати додатковий контроль виходу – наприклад, перевірка фактичних тверджень відповіді на відповідність знайденим джерелам або обмеження на формат відповіді. Це допомагає підтримувати точність та доречність відповіді.

Просунуті підходи: Naive vs Advanced RAG

- **Наївний RAG:** базова реалізація RAG-пайплайну. Містить мінімальний набір кроків: індексація даних, простий пошук (наприклад, топ--K найбільш схожих фрагментів) та одноетапна генерація відповіді. Немає додаткової оптимізації – система прямо використовує знайдені дані як є.

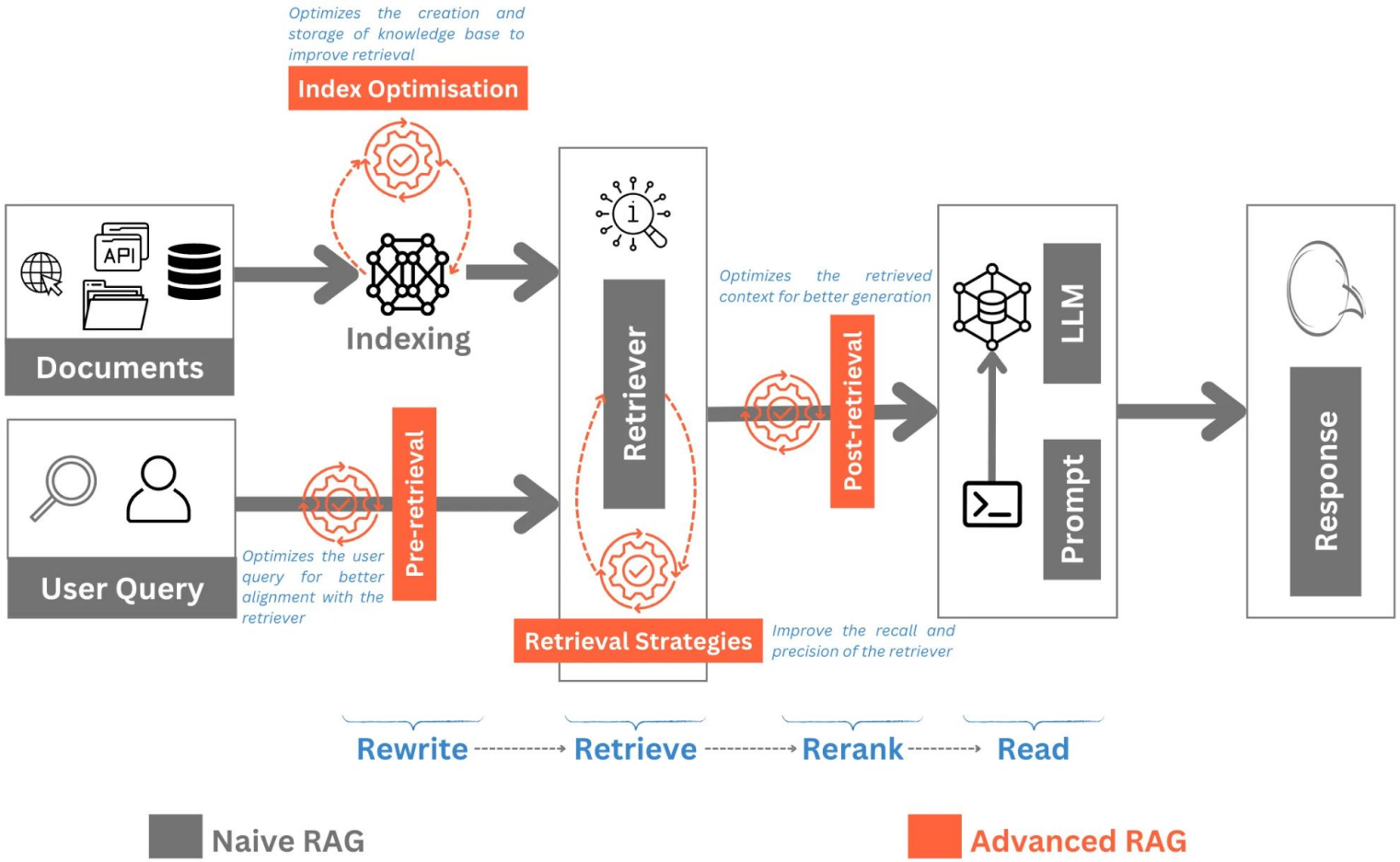
- **Просунутий RAG:** вдосконалений підхід, що включає додаткові кроки перед і після пошуку для підвищення якості. Дозволяє краще впоратися зі складними запитами та недоліками базового підходу.

- **Переписування запиту:** перед пошуком система може автоматично уточнювати або перефразовувати запит, щоб отримати більш релевантні результати. Наприклад, додати синоніми, контекст або розбити складне питання на підпитання.

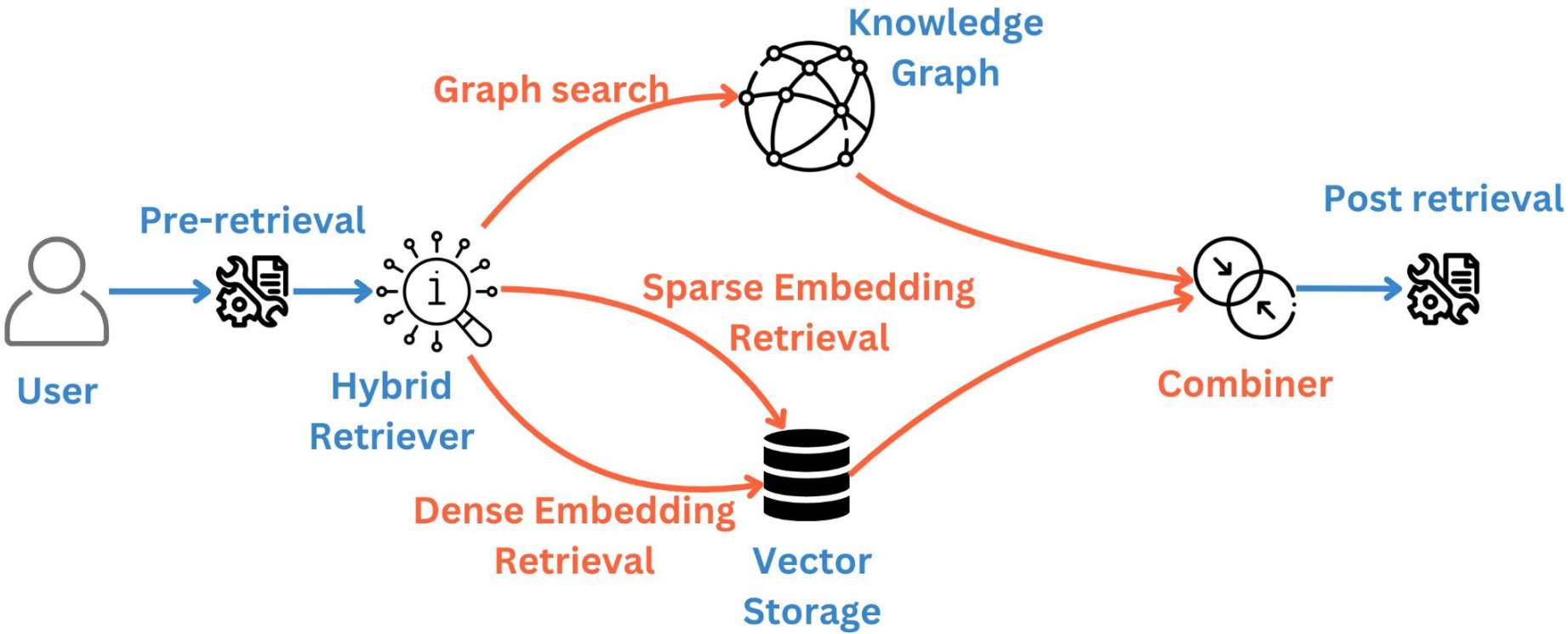
- **Реранжування та фільтрація:** після первинного пошуку отримані фрагменти можуть бути відфільтровані (видалення нерелевантних) та повторно відранжовані. Можуть використовуватися більш “важкі” моделі (наприклад, cross-encoder) для оцінки кожного кандидата і відбору найкращих перед подачею в LLM.

- **Ітеративна генерація:** у складних випадках просунутий RAG може виконувати кілька циклів: початкова відповідь LLM може перевірятися і на її основі робитися додатковий запит до бази знань (для уточнення) перед фінальною відповіддю. Це агентний підхід, що забезпечує досконаліший результат для комплексних задач.

Просунуті підходи: Naive vs Advanced RAG



Просунуті підходи: Graph RAG

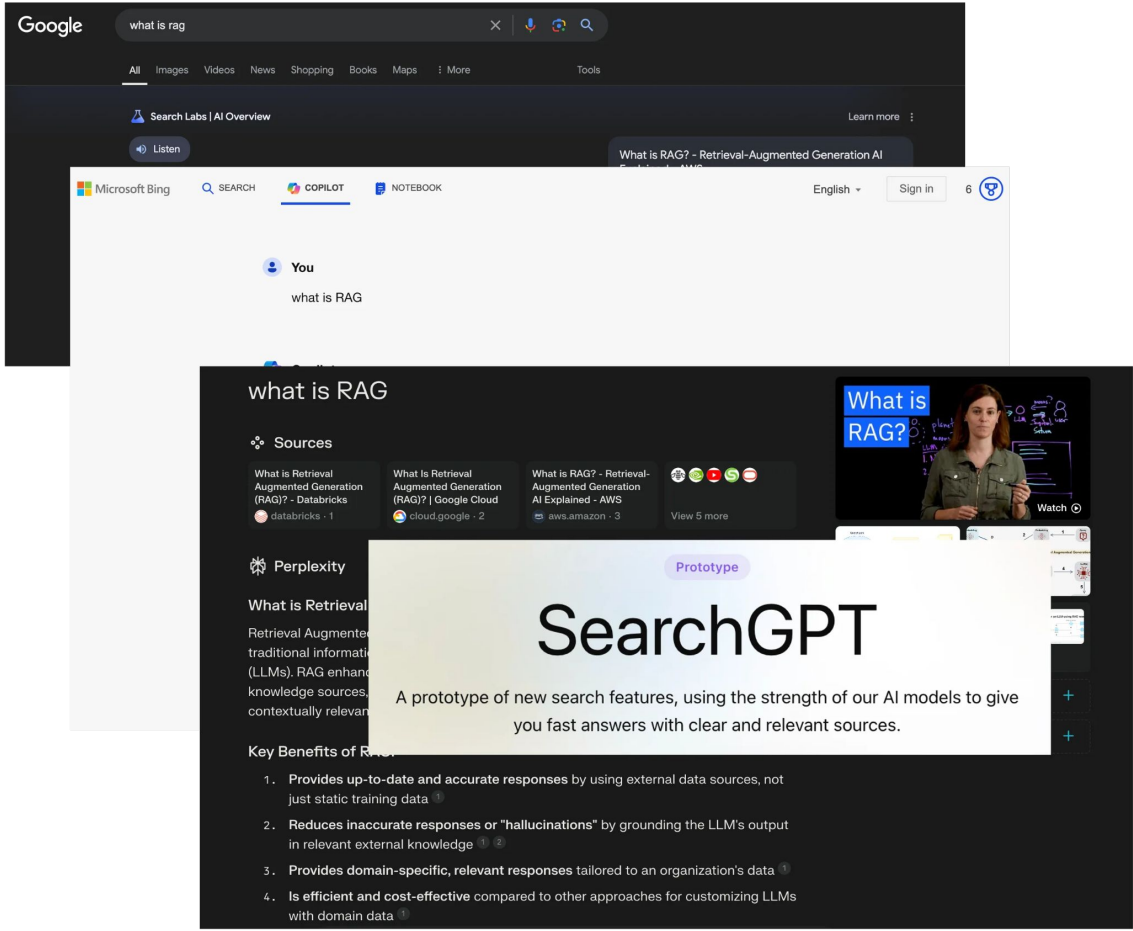


Оцінка якості RAG-систем

- **Precision та Recall:** метрики інформаційного пошуку. *Precision (точність)* – частка знайдених документів, що є релевантними запиту (вимірює наскільки “чисті” результати без зайвих). *Recall (повнота)* – частка релевантних документів, які система знайшла відносно всіх існуючих релевантних (чи не пропущено важливої інформації). Баланс цих показників важливий для ефективного пошуку.
- **F1-міра:** комбінована метрика якості пошуку, гармонійне середнє Precision і Recall. Використовується для загальної оцінки роботи пошукового компонента, особливо коли потрібно врахувати і точність, і повноту одночасно.
- **Релевантність та вірність відповіді:** якісні показники генерації. *Релевантність* – наскільки згенерована відповідь відповідає заданому запитанню та покриває його суть. *Вірність (фактична точність)* – наскільки відповідь базується на правдивій інформації з джерел, без вигаданих фактів. Вірність іноді називають *groundedness* – прив’язаністю відповіді до наданого контексту.
- **BEIR:** стандартний набір бенчмарків (Benchmarking IR) для оцінки якості пошуку по різних задачах і доменах. Наприклад, використовують BEIR для порівняння різних моделей embeddings – цей бенчмарк став “золотим стандартом” оцінювання якості пошуку у RAG-системах
- **Метрика RGB:** комплексна оцінка якості Retrieval-Augmented Generation по трьох напрямах – *R (Retrieval)*, *G (Generation)* та *B (Blend)*. Аналізується баланс між якістю пошуку і якістю згенерованої відповіді, а також їх сумарний ефект. Такий підхід дає цілісне уявлення про ефективність RAG: чи система знаходить потрібні дані і чи правильно їх використовує у відповіді.

Практичне застосування RAG

- **Медицина:** інтелектуальні медичні консультанти, що відповідають на питання лікарів або пацієнтів, спираючись на базу медичних знань.
- **Бізнес:** корпоративні чат-боти і довідкові системи..
- **Освіта:** персональні навчальні помічники, що допомагають студентам і учням.
- **Пошукові системи:** інтеграція RAG у веб-пошук для покращення відповідей.



Виклики при реалізації RAG

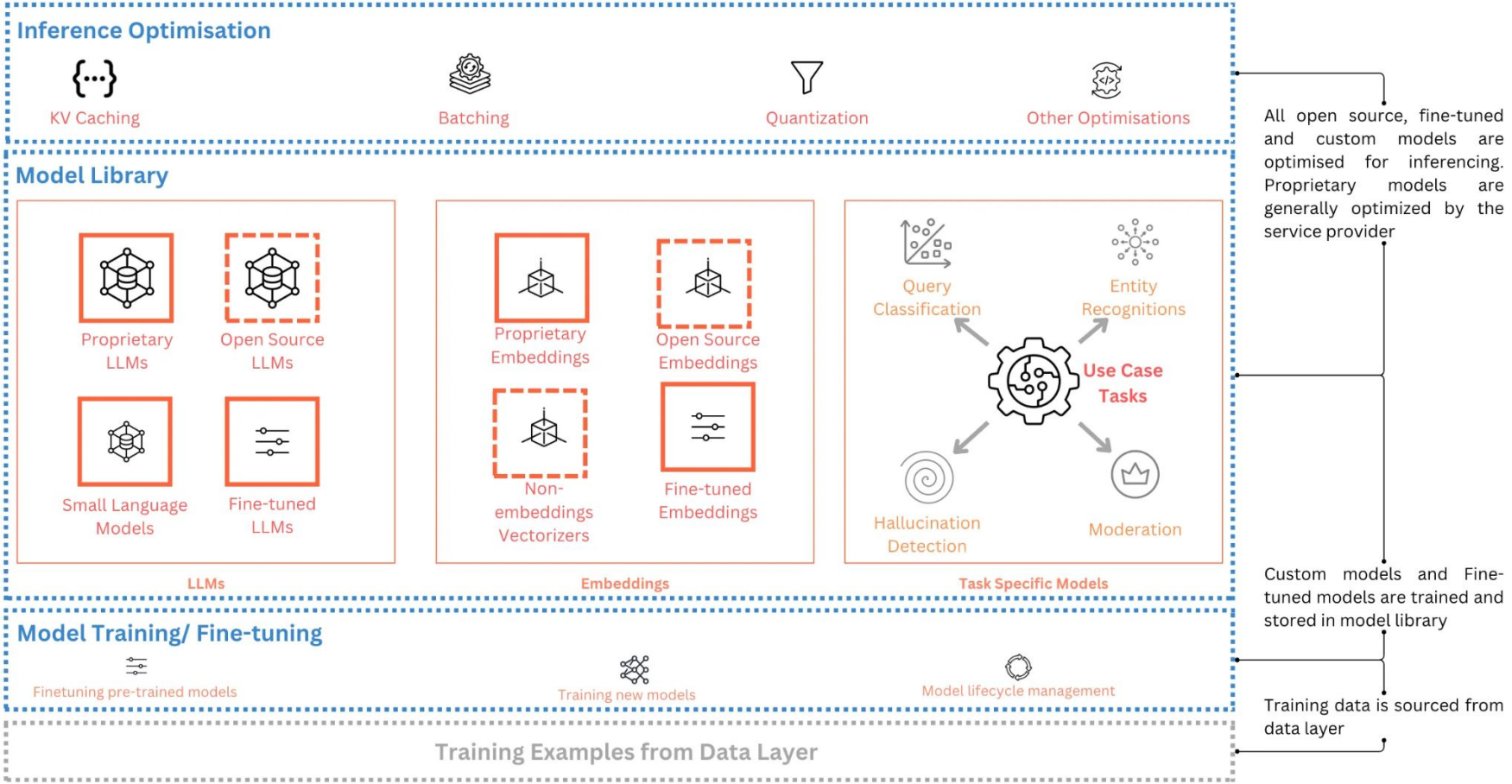
- **Технічні виклики:** масштабування системи під великий обсяг даних та запитів (потрібна оптимізація індексу для швидкого пошуку); обмеження розміру контексту LLM (всі знайдені дані можуть не поміститися у вікно вводу моделі); інтеграція різних компонент (налаштування сумісності між пошуком і генерацією, форматування даних для промпту).

- **Етичні виклики:** можливі упередження (bias) в модельних відповідях через упередженість у навчальних даних або у базі знань; ризик надання шкідливих порад або конфіденційної інформації, якщо база знань містить такі дані; необхідність контролю генерації, щоб уникнути токсичного або дискримінаційного контенту.

- **Приватність:** RAG-системи, що працюють з внутрішніми даними, повинні забезпечувати конфіденційність. Виклики включають безпечне зберігання приватних документів у базі знань (шифрування, контроль доступу) та недопущення витоку даних через відповіді. Якщо використовується стороння LLM API, виникають ризики передачі чутливої інформації третьим сторонам.

- **Витрати:** підтримка RAG може бути ресурсомісткою. Великі LLM вимагають значних обчислювальних ресурсів (GPU/TPU) під час генерації; зберігання та пошук по векторній базі – теж не безкоштовні (пам’ять, процесорний час). До того ж, регулярне оновлення бази знань і перенавчання embedding-моделей під нові дані потребує додаткових затрат часу та коштів.

Рівень моделі стека RAGOps



Перспективи розвитку RAG

- **Knowledge Graph RAG:** використання графів знань (семантичних мереж) у RAG. Замість неструктурованого тексту, система оперує структурованими знаннями: вузлами (сутностями) та зв'язками. Це дозволяє робити семантичний пошук за зв'язками між поняттями і покращує логічну узгодженість відповіді. Перспектива: інтеграція RAG з базами знань типу Wikidata, онтологіями в доменних системах для точніших відповідей на складні запити.
- **Multimodal RAG:** розширення принципів RAG на різні типи даних. Мультимодальні RAG-системи можуть працювати не лише з текстом, а й з зображеннями, аудіо, відео. *Наприклад:* питання користувача може супроводжуватися зображенням – система витягує інформацію як з текстової бази, так і аналізує зображення (через CV-модель), і генерує відповідь, поєднуючи обидва джерела. Це відкриває шлях до застосувань у сфері відеоархівів, доповненої реальності, аналізу даних з різних модальностей.
- **Agentic RAG:** поєднання RAG з агентним підходом. LLM-агенти можуть самостійно планувати послідовність дій для виконання завдання: формувати проміжні запити, виконувати кілька циклів пошуку, викликати додаткові інструменти (наприклад, калькулятор, зовнішні API). Agentic RAG означає, що модель не просто пасивно відповідає, а й активно вирішує як отримати інформацію. Перспектива – створення систем, здатних до багатокрокового міркування та діалогу з користувачем, наприклад, помічники, які можуть уточнювати запитання, знаходити кілька джерел, перевіряти факти і виводити підсумок.
- **Подальший розвиток:** RAG-технології будуть удосконалюватися в напрямку більшої швидкодії, точності та інтеграції з різними форматами даних. Можна очікувати появи нових варіантів RAG (наприклад, *Speculative RAG* для пришвидшення генерації, *Corrective RAG* з автоматичним виправленням помилок відповіді). У комплексі, RAG залишатиметься ключовою технологією для створення **пояснюваних** та **актуальних** AI-систем, що поєднують переваги великих моделей і перевірених знань..

Thank you