Практичне завдання №9: Проектування та аналіз систем Retrieval-Augmented Generation (RAG)

Мета: Зрозуміти архітектуру та принципи роботи RAG-систем, навчитись проектувати їх для конкретних предметних областей, порівнювати RAG з Fine-Tuning та аналізувати ключові елементи промптів для RAG.

Ключові концепції: Retrieval-Augmented Generation (RAG), база знань, чанкінг (chunking), векторизація, пошук за схожістю, гібридні пошуки, галюцинації, Fine-Tuning.

Завдання 1: Концептуальне проектування RAG-системи для предметної області

Ідея: Застосувати знання про RAG для розробки системи в контексті власної професійної або навчальної сфери.

Завдання: Розробіть концепцію RAG-системи, яка б виступала в ролі «помічника» у вашій фаховій предметній області або такій, що для вас представляє інтерес (наприклад, помічник юриста по кодексах, помічник лікаря по протоколах лікування, помічник інженера по технічній документації, помічник маркетолога по аналітичних звітах тощо).

Питання для роздумів, планування та опису в звіті (до 1 сторінки):

1. **Назва та призначення RAG-системи**: Дайте назву вашій системі та чітко сформулюйте, яку проблему вона вирішуватиме.
2. **Джерела знань (База Знань)**:
   - Які конкретні типи документів або даних стали б вашою "базою знань"? (Наприклад, для юриста: Кримінальний кодекс України, Цивільний кодекс, судові прецеденти).
   - Які формати даних ви б підтримували? (PDF, DOCX, HTML, JSON, бази даних).
   - Звідки б надходила ця інформація? (Публічні реєстри, корпоративні бази даних, внутрішні звіти).
3. **Стратегія чанкінгу та векторизації**:
   - Як би ви розбивали ваші документи на «чанки» (логічні фрагменти) для подальшої векторизації та пошуку? (Наприклад, по статтях кодексу, по розділах медичних протоколів, по параграфах технічної документації тощо).
   - Якого розміру були б ці чанки (орієнтовно кількість слів/токенів)?

- Які «метадані» (автор, дата, джерело тощо) ви б зберігали разом з кожним чанком?

4. **Приклади запитів**:
   - Наведіть 3 приклади реалістичних запитів, які користувач міг би поставити вашій RAG-системі.
   - Поясніть, чому стандартний LLM (без RAG) не впорався б з цими запитами або дав би неточну відповідь, а ваша RAG-система — впоралася б завдяки доступу до конкретних джерел.

Оцініть:

1. Наскільки глибоко розумієте, які джерела знань є релевантними для вашої предметної області.
2. Чи обґрунтована ваша стратегія чанкінгу? (Наприклад, розбиття по сторінках для кодексів може бути неефективним, якщо стаття розбита між сторінками).
3. Чи дійсно запропоновані запити є тими, де RAG демонструє свою перевагу?

Завдання 2: RAG **проти** Fine-Tuning (**порівняльний аналіз**)

Ідея: Систематизувати розуміння відмінностей між RAG та Fine-Tuning за ключовими критеріями.

Завдання: Складіть порівняльну таблицю (у форматі Markdown) між RAG та Fine-Tuning за наведеними критеріями. У колонках опишіть, як кожна технологія поводиться за цим критерієм.

Критерії для порівняння та формат таблиці (**для заповнення**):

Критерій	RAG	Fine-Tuning
Вартість оновлення знань		
Ризик галюцинацій		
Конфіденційність даних		
Вимоги до обчислень		
Гнучкість у роботі з новими/змінюваними даними		
Здатність до глибокого розуміння нового контексту		

Оцініть:

1. Наскільки точно ви розрізняєте переваги та недоліки кожного підходу.
2. В яких сценаріях RAG є кращим вибором, а в яких Fine-Tuning?

Завдання для СРС: **Аналіз промпта для** RAG **та** «**Вибивання з рейок**»

Ідея: Дослідити роль промпта в RAG-системі, особливо інструкцій щодо поведінки моделі у випадку відсутності інформації та забезпечення фактологічної коректності RAG-систем.

Завдання:

1. **Проаналізуйте промпт**:
   - Поверніться до типового промпта для RAG-системи, де контекст для відповіді надається моделі.
   - Особливо зверніть увагу на таку інструкцію: «*Використовуй тільки наданий контекст для відповіді. Якщо відповіді в контексті немає, скажи: 'На жаль, я не можу знайти цю інформацію в наданих джерелах.' Не вигадуй інформацію.*»
   - Поясніть, чому, на вашу думку, ця інструкція є критично важливою для RAG-системи.
2. **Експеримент** «**Вибивання з рейок**»:
   - Сформулюйте промпт для LLM (будь-якої, до якої маєте доступ, напр., ChatGPT, Gemini).
   - Надайте моделі **якийсь непов'язаний або хибний** «**контекст**» (1-2 короткі абзаци тексту на абсолютно випадкову тему).
   - **Завдання** 1 (**з інструкцією**):
     - Запитайте у моделі щось, що неможливо знайти у наданому вами «контексті», використовуючи наведену вище інструкцію: *«Використовуй тільки наданий контекст для відповіді. Якщо відповіді в контексті немає, скажи: 'На жаль, я не можу знайти цю інформацію в наданих джерелах.' Не вигадуй інформацію. Контекст: [ваш непов'язаний або хибний контекст] Запитання: [ваше запитання, відповіді на яке немає в контексті]»*
     - Зафіксуйте відповідь моделі.

- **Завдання** 2 (**без інструкції**):
    - Повторіть той самий запит, але приберіть інструкцію: *«Якщо відповіді в контексті немає, скажи: 'На жаль, я не можу знайти цю інформацію в наданих джерелах.' Не вигадуй інформацію.»*, залиште *«Використовуй тільки наданий контекст для відповіді. Не вигадуй інформацію. Контекст: [ваш непов'язаний або хибний контекст] Запитання: [ваше запитання, відповіді на яке немає в контексті]»*
    - Зафіксуйте відповідь моделі.

3. **Напишіть звіт** (1-2 **сторінки**):
    - Чітко поясніть важливість інструкції про відсутність інформації в контексті.
    - Опишіть ваш експеримент: який «контекст» ви надали, яке запитання поставили.
    - Наведіть отримані відповіді для Завдання 1 та Завдання 2.
    - **Проаналізуйте**:
        - Що сталося, коли інструкцію прибрали?
        - Чи почала модель «галюцинувати» або вигадувати інформацію, яка не була в контексті?
        - Які ризики це несе для надійності RAG-системи?