Zadanie Rekrutacyjne: System QA z użyciem RAG (Retrieval-Augmented Generation)

Cel zadania

Twoim celem jest zbudowanie prostego systemu pytanie-odpowiedź (QA), który:

1. Odbiera pytanie od użytkownika,

2. Przeszukuje bazę dostarczonych dokumentów tekstowych,

3. Przekazuje najbardziej pasujące fragmenty do modelu językowego (LLM),

4. Generuje odpowiedź na podstawie znalezionych informacji.

To tzw. Retrieval-Augmented Generation (RAG) – podejście, które łączy duże modele językowe z własną bazą wiedzy.

Czas realizacji

Zadanie nie ma limitu czasowego, a samo w sobie nie powinno zająć więcej niż kilka godzin – możesz je wykonać we własnym tempie. Zależy nam na jakości rozwiązania i przemyślanym podejściu.

Technologie i środowisko

Zadanie wykonaj w Pythonie 3.10+.
Możesz użyć dowolnego sposobu uruchomienia modelu LLM, pod warunkiem że działa lokalnie (bez zewnętrznych API).

Polecane:
- HuggingFace Transformers
- Ollama
- llama.cpp

Niedozwolone:
- API OpenAI, Claude, Gemini itp.

Polecane biblioteki:
- sentence-transformers
- scikit-learn
- transformers (Hugging Face)
- numpy, pandas, tqdm

Struktura zadania

1. Baza wiedzy: folder docs/ z dostarczonymi przez nas 4 plikami.

2. Retrieval: embeddingi SentenceTransformers + cosine similarity.

3. Generacja odpowiedzi: model LLM z kontekstem.

4. Prezentacja wyników: pokaż odpowiedzi na zadane pytania.

Co dostarczyć

Struktura projektu:

rag_project/
├─docs/
├─rag_pipeline.py lub rag_pipeline.ipynb
└── README.md

To podstawowe wymagane pliki, jak kandydat będzie chciał / potrzebował stworzyć dodatkowe pliki czy foldery do wykonania zadania, jest to jak najbardziej akceptowalne.

README powinien zawierać:
- Jak uruchomić projekt,
- Jakie modele zostały użyte,
- Jakie pytania zadano,
- Najciekawsze obserwacje (np. jakość odpowiedzi, błędy, ograniczenia).

Obowiązkowe pytania do zadania

System QA powinien odpowiedzieć na następujące pytania:

1. Jakie modele LLaMa są dostępne?

2. Kto stworzył PLLuM?

3. Jaki model najlepiej działa na GPU z 24 GB VRAM?

Dodatkowo: przygotuj 2–3 własne pytania na bazie dokumentów w folderze `docs/`. Chcemy zobaczyć, jak projektujesz zapytania i testujesz system.

Kryteria oceny

1. Poprawność działania pipeline RAG

2. Trafność i spójność wygenerowanych odpowiedzi

3. Umiejętność doboru i uruchomienia modelu LLM

4. Zrozumienie embeddingów i similarity

5. Czytelność kodu i struktura projektu

6. Jakość dokumentacji i obserwacji w README

Sposób oddania zadania

W celu oddania zadania przygotuj archiwum `.zip` zawierające cały projekt.

Zawartość:
- Pełny kod źródłowy,
- Folder `docs/` z dokumentami źródłowymi,
- Plik `README.md` zawierający:
  • instrukcję uruchomienia,
  • użyte modele,
  • pytania i odpowiedzi,
  • najciekawsze obserwacje (jakość, błędy, ograniczenia),
- Krótkie nagranie wideo (np. OBS), prezentujące działanie systemu QA.
  Może to być zwykły screen recording bez narracji.

Nazwij plik ZIP w formacie: `Imie_Nazwisko_RAG_QA.zip` i prześlij go zgodnie z instrukcjami rekrutera.