ارزیابی کارشناس هوش مصنوعی | مانا افراز خاور

توضیحات پروژه:

هدف این تسک، ساخت یک سیستم پایه Retrieval-Augmented Generation (RAG) است که متون ورودی را به Embedding Vectors تبدیل کرده، در یک پایگاه دادهٔ برداری نگه‌داری می‌کند و هنگام دریافت پرسش کاربر، با جستجوی شباهت (Similarity Search) و تولید پاسخ توسط LLM، نتیجهٔ نهایی را به‌همراه منابع ارائه می‌دهد.

مواردی که لازم است انجام شود:

1. **انتخاب و توجیه پایگاه دادهٔ برداری**
   - استفاده از یک Vector DB.
   - مستندسازی دلیل انتخاب بر اساس:
     کارایی در جستجوی شباهت و درج داده،
     مقیاس‌پذیری و شیوهٔ استقرار،
     پشتیبانی جامعهٔ کاربری و ابزارهای توسعه.
2. **مدل‌سازی داده**
   - طراحی اسکیمای ذخیره‌سازی شامل:
     Document Text (متن بخش/چانک)،
     Embedding Vector،
     Metadata (شناسهٔ سند، منبع، تاریخ، برچسب موضوعی و ...).
   - توضیح چرایی این ساختار و اینکه چگونه به فیلترینگ بهتر، بهبود سرعت بازیابی و ارتقای کیفیت RAG کمک می‌کند.

3. **منطق سیستم RAG**
   - ***Ingestion Pipeline***

     تقسیم متن به چانک‌های مناسب (طول/همپوشانی مشخص)،

     تولید Embedding با مدل سبک (مثلاً all-MiniLM-L6-v2)،

     ذخیرهٔ بردار + متادیتا در پایگاه دادهٔ برداری

     امکان قرار دادن PDF در یک پوشه مشخص
   - ***Query Pipeline***

     دریافت سؤال و تولید Embedding،

     بازیابی k چانک مشابه (با امکان فیلتر بر اساس متادیتا)،

     ساخت پرامپت حاوی دستورالعمل‌ها + زمینهٔ بازیابی‌شده،

     ارسال به LLM و تولید پاسخ نهایی به‌همراه منابع.

     جداسازی کامل لایهٔ RAG از LLM و استفاده از یک API رایگان یا کم‌هزینه (مثلاً JabirLLM یا Gemini Free) برای تولید پاسخ؛ به‌نحوی که LLM فقط بر پایهٔ context بازیابی‌شده پاسخ بدهد و از خروج مستقیم محتوای دیتابیس بدون RAG جلوگیری شود.

4. **پیاده‌سازی API با FastAPI (بخش اختیاری و امتیازی)**
   - POST /ingest

     ورودی: یک سند یا مجموعه‌ای از متون.

     عملیات: Chunking، تولید Embedding (مثلاً با sentence-transformers/all-MiniLM-L6-v2 یا هر مدل دلخواه) و ذخیره در Vector DB.

     خروجی: تأیید ثبت + خلاصهٔ تعداد چانک‌ها/توکن‌ها/ابعاد بردار و شناسه‌ها.
   - POST /query

     ورودی: پرسش کاربر (+ پارامترهای اختیاری مثل k، فیلتر متادیتا).

     عملیات: تبدیل پرسش به بردار، Similarity Search، انتخاب k نتیجهٔ برتر، ساخت پرامپت ترکیبی و ارسال به LLM.

     خروجی: پاسخ نهایی + فهرست context chunks (منابع، امتیاز شباهت).
   - مستندسازی خودکار API با Swagger/Redoc (FastAPI docs).

5. **کیفیت و مستندات**
   - ثبت تنظیمات کلیدی (ابعاد بردار، k، آستانهٔ شباهت، طول چانک/همپوشانی).
   - توضیح طراحی پرامپت برای کاهش hallucination و وادار کردن مدل به استناد به منابع.
   - راهنمای اجرا (README): نحوهٔ راه‌اندازی، متغیرهای محیطی، نمونهٔ درخواست/پاسخ.

نکات کلیدی و موارد مهم

1- زمان انجام تسک: از زمانی که این تسک دریافت می‌شود، 72 ساعت فرصت دارید تا تسک را انجام داده و ارسال کنید.

2- محیط اجرا: پروژه را به‌صورت پابلیک در Github منتشر کنید.

• مخزن باید شامل موارد زیر باشد:

◦ کد کامل

◦ فایل requirements.txt یا environment.yml

◦ مستندات API (در صورت پیاده سازی)

◦ فایل README.md.

3- کیفیت کد و گزارش:

• کد تمیز، خوانا، مستند و قابل اجرا باشد.

• نتایج و شیوهٔ ارائه به شکل مرتب و شفاف بیان شود.

4- مصاحبهٔ فنی آنلاین: در جلسهٔ مصاحبهٔ فنی، ممکن است دربارهٔ کد و نحوهٔ پیاده‌سازی شما سؤال شود؛ آمادهٔ توضیح منطق و روش کارتان باشید.

5- ارتباط با ما: در صورت داشتن هرگونه سؤال یا ابهام، می‌توانید از طریق همین ایمیل با ما در ارتباط باشید.

توضیحات اضافی

• می‌توانید برای فیلترینگ متادیتا (مانند تاریخ/منبع/برچسب) در مرحلهٔ بازیابی، گزینه‌های اختیاری API قرار دهید.

• توجه به حریم خصوصی: اطمینان از عدم ارسال دادهٔ حساس به سرویس‌های بیرونی (در صورت نیاز به سانسور/ناشناس‌سازی).

• تمامی بخش‌های پیاده‌سازی‌شده با توابع درج شود.