HomeWork

Generative AI

RAG and chain

1. فایل تمرین را در پنل خود آپلود کنید.
2. title فایل تمرین به صورت (نام تمرین+نام و نام خانوادگی) به انگلیسی باشد.
3. در صورتی که سوال و یا ابهامی دارید در گروه چت تلگرامی بپرسید.
4. فایل های پیوست را می‌توانید از اینجا دریافت کنید.

۱. ایجاد چت‌بات بر روی داده‌های شخصی با استفاده از RAG

در این تمرین، با استفاده از مفاهیمی که در کلاس آموخته‌اید، یک retriever برای داده‌های محلی خود پیاده‌سازی خواهید کرد. سپس با بهره‌گیری از مفهوم chain، یک چت‌بات اختصاصی طراحی می‌کنید که قادر به پاسخ‌گویی به سؤالات مرتبط با اسناد خواهد بود.

مرحله ۱: بارگذاری داده‌ها

در فایل پیوست، فهرستی از URLهای مرتبط با آیین‌نامه‌های بانک مرکزی در اختیار شما قرار گرفته است. برای بارگذاری این اسناد در محیط کد خود، از dataloader های مختلفی که در کلاس معرفی شده‌اند، استفاده کنید. می‌توانید:

- اسناد را مستقیماً بارگذاری کرده و به متن تبدیل کنید، یا
- ابتدا آن‌ها را به فرمت PDF دانلود کرده و سپس با ابزارهای PDF loader متنشان را استخراج نمایید.

برای جلوگیری از اجرای مجدد این فرآیند در هر بار اجرا، پیشنهاد می‌شود که متن استخراج‌شده اسناد را ذخیره کنید تا در دفعات بعدی تنها متن پردازش‌شده را بارگذاری نمایید. (در صورت تمایل، می‌توانید از متون دیگری نیز استفاده کنید.)

مرحله ۲: ایجاد RAG

در این بخش، باید از داده‌هایی که در مرحله قبل بارگذاری کرده‌اید، یک پایگاه داده برداری (Vector Database) بسازید تا چت‌بات شما قابلیت جستجوی معنایی و بازیابی اسناد مرتبط را داشته باشد.

برای پیاده‌سازی retriever چندین گزینه پیش روی شماست:

- در ساده‌ترین حالت، می‌توانید با استفاده از کتابخانه FAISS یک پایگاه داده از اسناد خود بسازید و سپس با متد as_retriever() آن را مستقیماً به retrieverتبدیل کنید. توجه: اسناد طولانی را به chunkهای کوچک‌تر تقسیم کنید.
- همچنین می‌توانید از روش‌های پیشرفته‌تر مانند ParentDocumentRetriever یا ترکیب روش‌های embedding search با تکنیک‌های کلاسیک نظیر TF-IDF (که در کلاس آموخته‌اید) استفاده کنید تا دقت بازیابی اسناد افزایش یابد.
- برای جلوگیری از اجرای مجدد فرآیند embedding در هر بار اجرا، متدهای ذخیره و بازخوانی retrieverرا نیز پیاده‌سازی کنید.

مرحله ۳: ایجاد chain

در این مرحله باید یک chain مطابق با ساختار زیر ایجاد کنید.

```
RAG = RunnableParallel(
    {
        "context": itemgetter("question") | retriever,
        "question": RunnablePassthrough(),
    }
)
chain = RAG | prompt | llm | output_parser
```

مکانیزم کارکرد:

1. ورودی کاربر ابتدا به retriever ارسال می‌شود تا متن مرتبط از پایگاه داده بازیابی شود.
2. متن بازیابی‌شده همراه با سؤال کاربر در پرامپت طراحی‌شده قرار گرفته و به مدل زبانی (LLM) ارسال می‌شود.
3. خروجی مدل زبانی با استفاده از parser پردازش شده و در قالبی خوانا و بدون اطلاعات اضافی به کاربر نمایش داده می‌شود.

.

نحوه‌ی تحویل:

کد خود به همراه خروجی های ایجاد شده را به فرمت یک فایل ipynb. ارسال کنید.