



Home Work

Generative AI

RAG and chain

۱. فایل تمرین را در پنل خود آپلود کنید.



۲. فایل تمرین به صورت (نام تمرین+نام و نام خانوادگی) به انگلیسی باشد.

۳. در صورتی که سوال و یا ابهامی دارید در گروه چت تلگرامی بپرسید.

۴. فایل های پیوست را می توانید از [اینجا](#) دریافت کنید.

۱. ایجاد چت بات بر روی داده های شخصی با استفاده از RAG

در این تمرین، با استفاده از مفاهیمی که در کلاس آموخته اید، یک retriever برای داده های محلی خود پیاده سازی خواهید کرد. سپس با بهره گیری از مفهوم chain، یک چت بات اختصاصی طراحی می کنید که قادر به پاسخ گویی به سوالات مرتبط با اسناد خواهد بود.

مرحله ۱: بارگذاری داده ها

در فایل پیوست، فهرستی از URL های مرتبط با آیین نامه های بانک مرکزی در اختیار شما قرار گرفته است. برای بارگذاری این اسناد در محیط کد خود، از dataloader های مختلفی که در کلاس معرفی شده اند، استفاده کنید. می توانید:

-

استناد 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 استخراج شده استناد را ذخیره کنید تا در دفعات بعدی تنها متن پردازش شده 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 استناد 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 استناد خود بسازید و سپس با متدهای as_retriever() آن را مستقیماً به retriever تبدیل کنید. توجه: استناد 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 دقیق‌تر بازیابی استناد افزایش یابد.

-

برای جلوگیری از اجرای مجدد فرآیند embedding در هر بار اجرا، متدهای ذخیره و بازخوانی retriever را 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 بازیابی شده همراه با سؤال کاربر در پرامپت طراحی شده 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.

نحوهٔ تحويل:

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