Home Work

Generative AI

Ollama and embeddings

1. فایل تمرین را در پنل خود آپلود کنید.
2. فایل تمرین به صورت (نام تمرین+نام و نام خانوادگی) به انگلیسی باشد. title
3. در صورتی که سوال و یا ابهامی دارید در گروه چت تلگرامی بپرسید.
4. برای هر یک از قسمت‌های تمرین، یک فایل ipynb. به همراه خروجی اجرا شده‌ی هر cell را در فایل ارسالی قرار دهید.

۱. راه اندازی Ollama بر روی google colab

در این تمرین، شما مدل زبانی متن باز llama3.1 با هفت میلیارد پارامتر را از طریق ollama در محیط google colab سرو می‌کنید و سپس با استفاده از کتابخانه‌ی langchain-ollama یک instance از مدل زبانی ایجاد کرده و پرامپ خود را جهت اجرا به این مدل زبانی ارسال می‌کنید.

مرحله ۱: نصب و راه اندازی ollama در colab

در محیط colab به صورت پیش فرض به ران تایم CPU متصل می‌شوید ولی در مواردی که سرورهای GPU در دسترس

باشند می‌توانید به صورت محدود از GPU ارائه شده نیز استفاده کنید. در ابتدا سعی کنید از منوی Runtime و change runtime type تنظیمات را بر روی T4 GPU قرار دهید. در صورت اتصال به سرور GPU، با اجرای دستور زیر، پکیج های لازم جهت اجرای ollama بر روی GPU را نصب کنید:

```
!apt update && apt install -y pciutils lshw
```

با اجرای دستور زیر میتوانید از صحت اتصال به GPU اطمینان حاصل کنید:

```
!nvidia-smi
```

در صورتی که از CPU استفاده می‌کنید نیازی به اجرای این دستورات نیست و مستقیما مراحل زیر را اجرا کنید.

برای نصب ollama بر روی colab از دستور زیر استفاده کنید:

```
!curl -fsSL https://ollama.com/install.sh | sh
```

حال باید ollama را سرو کرده تا بتوانید از طریق API به آن متصل شوید. توجه داشته باشید که چون در محیط colab هستید، باید ollama را در background سرو کرده تا به شما اجازه‌ی اجرای باقی d سرو کرده تا به شما اجازه‌ی اجرای باقی cell ها داده شود. به این منظور از دستور زیر استفاده کنید:

```
!nohup ollama serve &
```

در ادامه با دستور زیر مدل زبانی را دانلود کنید:

```
!ollama run llama3.1
```

بعد از اتمام دانلود مدل زبانی، cell مورد نظر در حالت اجرا باقی می‌ماند. در این حالت عبارت bye/ را وارد کنید تا اجرای cell متوقف شود و بتوانید باقی cell ها را اجرا کنید.

نصب و راه اندازی ollama در محیط colab به شما این امکان را می‌دهد که بتوانید مدل‌های زبانی مختلف را به راحتی دانلود کرده و عملکرد آن ها را مورد بررسی قرار دهید.

مرحله ۲: اتصال به سرور ollama

حال که ollama را در محیط کولب اجرا کرده‌اید، با استفاده از کتابخانه‌ی langchain_ollama و کلاس ChatOllama یک instance از مدل زبانی سرو شده ایجاد کنید و عملکرد آن را با ارسال پرامپ های مختلف (فارسی و انگلیسی) مورد بررسی قرار دهید.

۲. استفاده از مدل‌های embedding به صورت لوکال

در این تمرین، شما یک vectorstore را با استفاده از faiss یک مدل embedding که قابلیت اجرا به صورت لوکال را دارد بر روی متون فارسی پیاده سازی می‌کنید.

بخش اول: دانلود مدل embedding

به این منظور، با استفاده از کتابخانه‌ی langchain_huggingface مدل متن باز parsbert را که برای embedding متون فارسی فاین تیون شده است را دانلود کنید. به این منظور می‌توانید بعد از نصب کتابخانه‌های مورد نیاز، از کد زیر در محیط گوگل کولب استفاده کنید (توجه داشته باشید که در صورت اجرا بر روی لپتاپ، نیازمند دانلود فایل های مدل با حجم حدود یک گیگابایت هستید):

```
!pip install langchain langchain-huggingface langchain_community faiss-cpu
```

```
from langchain_huggingface.embeddings import HuggingFaceEmbeddings


model_name = "HooshvareLab/bert-base-parsbert-uncased"

model_kwargs = {'device': 'cpu'}
encode_kwargs = {'normalize_embeddings': False}
hf_embedding = HuggingFaceEmbeddings(
model_name=model_name,
model_kwargs=model_kwargs,
encode_kwargs=encode_kwargs
)
```

حال برای تست عملکرد مدل، می‌توانید کد زیر را اجرا کنید:

```
text = " ما در هوشواره معتقدیم با انتقال صحیح دانش و آگاهی، همه افراد
می‌توانند از ابزارهای هوشمند استفاده کنند. شعار ما هوش مصنوعی برای همه
است."

embed = hf_embedding.embed_query(text)
print(len(embed))
```

بخش دوم: ساخت vectorstore

در این مرحله، با استفاده از مطالبی که در کلاس در مورد نحوه‌ی ایجاد vectorstore با استفاده از faiss آموخته‌اید و با استفاده از مدل embedding که به صورت لوکال ایجاد کرده اید، یک متن فارسی را با استفاده از chunk های مناسب به تعدادی document تبدیل کرده، و یک vectorstore که قابلیت جستجوی معنایی داشته باشد ایجاد کنید.

```
index = faiss.IndexFlatL2(len(hf_embedding.embed_query("سلام")))

vector_store = FAISS(
embedding_function=hf_embedding,
...
)
```

در انتها می‌توانید نتیجه‌ی عملکرد مدل embedding لوکال را با سایر مدل مدل ها نظیر openai مقایسه کنید.