به نام خدا

دانشگاه تهران
دانشکدگان فنی
دانشکده مهندسی برق و کامپیوتر

درس پردازش زبان طبیعی

تمرین شماره ۶

استاد درس: دکتر هشام فیلی

سرپرست دستیاران آموزشی: سمانه پیمانی‌راد

طراح تمرین: میلاد محمدی

دی ماه ۱۴۰۳

فهرست

مقدمه

موضوع تمرین:

در این تمرین، شما به طراحی و پیاده‌سازی یک چت‌بات به نام **چت‌فود** خواهید پرداخت. چت‌فود بخشی از یک اپلیکیشن فرضی سفارش غذا است که هدف آن ارائه‌ی خدمات به کاربران از جمله جستجو غذاها، ارائه اطلاعات عمومی و تخصصی درباره‌ی خوراکی‌ها و غذاها، پیشنهاد غذا و پیگیری سفارشات می‌باشد. در این پروژه، شما با استفاده از فریم‌ورک **LangGraph** یک چت‌بات هوشمند طراحی خواهید کرد که قابلیت پاسخ‌گویی به سوالات کاربران و اجرای سرویس‌های ساده مرتبط با پیگیری سفارش‌ها را داشته باشد. این تمرین فرصتی است تا شما به صورت عملی با فرآیند طراحی و توسعه‌ی چت‌بات‌ها آشنا شوید.

در فرآیند این تمرین، شما علاوه بر طراحی چت‌بات، با ابزارهایی مانند **LanceDB** برای ذخیره‌سازی و بازیابی اطلاعات، و همچنین ابزارهای تکمیلی مانند **Tavily** و **LlamaParse** کار خواهید کرد. برای هسته پردازش زبان طبیعی چت‌بات خود می‌توانید با انتخاب خودتان از بین مدل‌های بزرگ زبانی موجود مانند **Llama 3.3 70B**، **Gemini Flash 1.5**، **GPT4o-mini** و **Command R** استفاده کنید که API آن‌ها به از طریق وب‌سایت ارائه‌دهندگان در دسترس است. باتوجه به عدم امکان دسترسی رایگان به مدل‌های قوی‌تر در زبان فارسی مانند **GPT4o** زبان ارتباط با چت‌بات در تمام مراحل **انگلیسی** خواهد بود. (هم ورودی کاربر و هم خروجی انگلیسی باشد.)

در انتها، چت‌فود شما باید به یک رابط کاربری مناسب با استفاده از ابزار **Chainlit** متصل شود تا تجربه کاربری مطلوبی ارائه دهد. طی هر مرحله از تمرین، شما چت‌بات خود را تست و ارزیابی خواهید کرد تا اطمینان حاصل کنید که خروجی طراحی‌شده از نظر کارایی و تعامل‌پذیری، استانداردهای لازم را داراست.

اهداف تمرین:

- **آشنایی و کار با فریم‌ورک LangGraph**: دانشجویان با فریم‌ورک LangGraph آشنا می‌شوند و از آن برای طراحی و پیاده‌سازی یک چت‌بات هوشمند استفاده می‌کنند.
- **کار با ابزارهای مرتبط در ساخت چت‌بات و عامل‌های هوشمند**: دانشجویان از ابزارهایی مانند Tavily برای ایجاد امکان جستجو در اینترنت، LanceDB برای ذخیره‌سازی و بازیابی داده‌ها، LlamaParse برای پردازش فایل‌ها و Chainlit برای ایجاد رابط کاربری استفاده خواهند کرد.
- **توانایی طراحی کلی چت‌بات و عامل هوشمند براساس نیازمندی‌ها**: در این تمرین، دانشجویان مهارت طراحی معماری یک چت‌بات و عامل هوشمند را کسب می‌کنند و نحوه تطبیق طراحی خود با نیازمندی‌های تعریف‌شده را می‌آموزند.
- **مهارت مطالعه مستندات مربوط به ابزارهای برنامه‌نویسی**:
دانشجویان یاد می‌گیرند چگونه مستندات ابزارها و مدل‌های برنامه‌نویسی را مطالعه کنند و در مدت زمان کوتاهی آن‌ها را در کدهای خود پیاده‌سازی کنند.
- **بهبود مهارت ارزیابی و بهینه‌سازی چت‌بات‌ها**: دانشجویان در طول تمرین، چت‌بات خود را تست و ارزیابی می‌کنند تا خروجی طراحی‌شده از لحاظ کارایی و تعامل‌پذیری بهبود یابد.

نکات قابل توجه در هنگام پاسخ به سؤالات:

۱- فرمت کد و فایل‌ها

برای اتصال به رابط کاربری، اجرای نهایی تمرین باید در یک فایل با فرمت py. انجام شود. با این حال، برای ارائه توضیحات و پاسخ به سوالات، نیاز است نسخه‌ای از همان کد در قالب فایل ipynb. تهیه شود. این فایل نوت‌بوک باید شامل مستندسازی بخش به بخش کد به همراه توضیحات کلی کد و معماری، و پاسخ به سوالات مطرح‌شده در متن تمرین باشد. اطمینان حاصل فرمایید که سوالات داخل متن تمرین را در توضیحات سلول‌های متنی کد خودتان بیاورید. نیازی به ارائه جزئیات خط به خط کد نیست، اما توضیحات باید شفاف و کافی باشند تا روند طراحی و پیاده‌سازی چت‌بات به‌خوبی منعکس شود.

۲- ذخیره‌سازی logها

در طول اجرای تمرین، تمام مراحل گفتگو، پیام‌های رد و بدل شده، وضعیت چت‌بات (مانند استفاده از ابزارها و وضعیت یا همان state) باید ثبت و در قالب log ذخیره شوند. این logها باید در یک فایل مناسب برای چند سناریو مختلف چت ارائه شوند و همراه با کد تمرین ارسال گردند تا صحت و دقت عملکرد چت‌بات قابل ارزیابی باشد. نیازی به یک log پیشرفته نیست و صرفا نمایش کلی وضعیت چت‌بات هدف است، می‌توانید از langsmith نیز برای این هدف استفاده کنید. (صرفا لاگ مربوط به چند سناریو تست بر روی نسخه‌ی نهایی ربات خودتان را پیوست کنید و نسخه‌های آزمایش با هدف دیباگ کد را نفرستید.)

۳- فیلم ارائه تمرین

در انتهای تمرین، لازم است یک فیلم ۳ تا ۵ دقیقه‌ای از عملکرد چت‌بات خود تهیه کنید. این فیلم باید شامل نمایش تعامل با چت‌بات، توضیح قابلیت‌های آن، و مروری کلی بر معماری طراحی‌شده باشد. نیازی به ارائه جزئیات دقیق کد در این فیلم نیست، اما باید توانایی‌ها و ویژگی‌های اصلی چت‌بات به‌طور واضح نمایش داده شوند.

۴- طراحی معماری چت‌بات

این تمرین گام‌به‌گام طراحی نشده است و بخشی از نمره به توانایی شما در طراحی معماری چت‌بات اختصاص دارد. لازم است گراف و زیرگراف‌های بخش‌های مختلف چت‌بات خود را تهیه کرده و در گزارش تمرین قرار دهید. این گراف‌ها باید معماری کلی و نحوه ارتباط اجزای مختلف چت‌بات را نمایش دهند. توجه داشته باشید که شما یک چت‌بات ارائه می‌دهیم که همه قابلیت‌های بیان شده را دارد، نه آنکه چند نسخه‌ی متفاوت و مستقل داشته باشید.

۵- تست و مدیریت خطا

کد چت‌بات باید به گونه‌ای طراحی شود که در شرایط مختلف به درستی عمل کند و خطاها را مدیریت کند. کرش کردن کد، پاسخ‌های نادرست، یا گیر کردن در وضعیت‌های خاص از موارد عدم صحت تمرین است. چت‌بات باید بتواند در صورت نیاز کاربر را راهنمایی کرده و درخواست او را دوباره دریافت کند. حتماً سناریوهای مختلف را تست و ارزیابی کنید.

توصیه آخر...

با توجه به اینکه فریم‌ورک‌های مورد استفاده در این تمرین جدید هستند و منابع آموزشی محدودی برای آنها وجود دارد، توصیه می‌شود انجام این تمرین را به روزهای آخر موکول نکنید. استفاده سریع و موثر از مستندات ابزارها، یکی از اهداف این تمرین است و نیاز به زمان کافی برای مطالعه و پیاده‌سازی دارد .

چت‌فود (۱۰۰ نمره)

همان‌طور که در مقدمه اشاره شد، در این تمرین، شما چت‌باتی به نام چت‌فود طراحی و پیاده‌سازی خواهید کرد که بخشی از یک اپلیکیشن فرضی سفارش غذا است. این تمرین شما را با استفاده از مدل‌های زبانی بزرگ به عنوان هسته‌ی اصلی پردازش زبان و ابزارهای مدرن در طراحی چت‌بات آشنا می‌کند. در ادامه، بخش‌های اصلی تمرین به تفصیل توضیح داده شده است.

در این تمرین، از مدل‌های زبانی بزرگ برای پردازش زبان، استدلال قدم به قدم، تصمیم‌گیری، و تولید پاسخ‌های کاربرپسند استفاده می‌شود. شما نیازی به فاین‌تیون کردن یا تغییر وزن‌های مدل ندارید. در اولین قدم، لازم است مدلی را انتخاب کنید که برای کارهایی مانند استدلال منطقی، تصمیم‌گیری، و تولید پاسخ‌های متناسب مناسب باشد. مدل پیشنهادی ما Gemini 1.5 Flash است، اما می‌توانید از مدل‌های دیگر نیز استفاده کنید.

چت‌فود یک دستیار هوشمند است که به کاربران کمک می‌کند درباره‌ی غذاها و خوراکی‌ها اطلاعات کسب کنند، پیشنهادات غذایی دریافت کنند، و سفارش‌های قبلی خود را پیگیری کنند. این چت‌بات هیچ عملیات سفارش یا پرداختی انجام نمی‌دهد و تمرکز آن صرفاً بر ارائه خدمات دستیارانه است. در طراحی خود باید این مورد را در نظر داشته باشید که این چت‌فود جایگزین اپلیکیشن سفارش غذا نیست، اما یک دستیار خوب و کاربرپسند برای آن است.

کدی شامل دو جدول پایگاه داده (سفارشات و غذاها) و یک بک‌اند ساده برای این تمرین آماده شده است. شما نیازی به ارتباط API ندارید و می‌توانید به‌راحتی با قراردادن این کد و پایگاه‌داده در همان مسیر برنامه‌نویسی خود از این توابع استفاده کنید. توصیه می‌شود فایل db_manager.py را تغییر ندهید، اما در صورت لزوم می‌توانید با ارائه دلایل مستدل، تغییرات لازم را اعمال کنید. توضیحات دقیق توابع و جدول‌های پایگاه داده در بخش مربوطه آمده است.

(مثال‌ها در این قسمت و قسمت‌های دیگر به فارسی نوشته شده‌اند، اما توجه داشته باشید که زبان چت‌بات شما در دریافت پیام و پاسخ به طور کامل انگلیسی است.)

قابلیت‌های اصلی چت‌فود

۱. **پاسخ‌دهی به اطلاعات عمومی و تخصصی غذاها:**

در این بخش، از معماری **Agentic RAG/Hybrid RAG** استفاده خواهید کرد. چت‌فود باید به سوالات کاربران درباره‌ی خوراکی‌ها و غذاها پاسخ دهد. این سوالات می‌توانند درباره‌ی ماهیت یک غذا یا توصیه‌های تغذیه‌ای باشند، اما این سوالات به طور مستقیم شامل جستجو در منوی رستوران‌ها نمی‌شوند. به عنوان مثال:

- «کباب مراکشی چیست؟»
- «آیا خوردن ماست کنار کباب ضرر دارد؟»

۲. **خدمات مشتریان**

کاربران می‌توانند وضعیت سفارش‌های خود را پیگیری کنند، سفارش را لغو کنند، یا نظر خود را ثبت کنند. شما باید از توابع بک‌اند برای دسترسی به پایگاه داده سفارشات استفاده کنید. این بخش شامل سناریوهایی مانند:

- «سفارشم دستم نرسیده، میشه پیگیری کنی؟»
- «می‌خوام سفارش به شماره ۱۲۳ رو کنسل کنم.»

۳. **جستجوی غذا**

این قابلیت به کاربران اجازه می‌دهد با استفاده از زبان طبیعی غذاهای موجود در رستوران‌ها را جستجو کنند. هدف این بخش جایگزین کردن زبان طبیعی به‌جای نوار جستجو در اپلیکیشن‌های سفارش غذا است. (اما شامل ثبت سفارش نمی‌شود) به عنوان مثال:

- «چه رستوران‌هایی پیتزا دارند؟»
- «قیمت قورمه سبزی رستوران میلاد چنده؟»

٤. **پیشنهاد غذا**

این بخش از معماری Reflection Agent یا معماری‌های مشابه مانند Plan and Executeاستفاده می‌کند. چت‌فود باید ورودی کاربر را تحلیل کرده، غذاهای مرتبط را جستجو کند، موجود بودن آن‌ها را بررسی کند و سپس پاسخ مناسبی ارائه دهد. این پاسخ باید به نحوی باشد که به کاربر بگوید براساس ورودی آن پیشنهادی که برایش دارد مثلا فلان غذا است که این رستوران‌ها دارند. به عنوان مثال:

- «یه غذای فست‌فودی تند می‌خوام.»
- «حوس یه غذای عربی کردم»

۵. **رابط کاربری**

در انتها، چت‌فود باید در قالب یک رابط کاربری مناسب ارائه شود. پیشنهاد ما استفاده از **Chainlit** است که امکان اتصال سریع و ساده به UI را فراهم می‌کند. ابتدا تمامی قابلیت‌ها را در محیط برنامه‌نویسی خود (ترمینال) تست کنید و سپس رابط کاربری را اضافه کنید.

٦. **بخش‌های امتیازی**

بخش‌های امتیازی شامل توسعه قابلیت‌های پیشرفته‌تر در هر بخش است. این موارد نیازمند مطالعه دقیق‌تر مستندات ابزارها وقت‌گذاری بیشتر بر روی این تمرین است. جزئیات این بخش‌ها در انتهای تمرین آمده است.

در ادامه این بخش، ابزارهای مورد استفاده و کدهای پایگاه داده و بک‌اند به تفصیل توضیح داده خواهند شد. همچنین هرکدام از قابلیت‌های ذکر شده به‌صورت کامل شرح داده می‌شود. مطمئن شوید که تمامی بخش‌های تمرین را با دقت مطالعه و اجرا کنید.

توضیح ابزارها و فریم‌ورک‌های مورد استفاده

در این بخش توضیحات خلاصه‌ای درباره‌ی ابزارهای مورد استفاده در این تمرین آورده شده است. اگر برای هرکدام از کاربردهای مورد نظر از ابزار دیگری استفاده می‌کنید حتما باید از دستیار تمرین تایید بگیرید وگرنه کد شما مورد قبول نخواهد بود.

LangGraph

فریم‌ورک LangGraph برای توسعه‌ی عامل‌ها، چت‌بات‌ها و راه‌حل‌های مبتنی بر مدل‌های زبانی بزرگ طراحی شده است. این ابزار امکانات متنوعی را به صورت توابع آماده ارائه می‌دهد و پس از فریم‌ورک LangChain معرفی شده است، اما همچنان از برخی ماژول‌ها و رابط‌های آن استفاده می‌کند LangGraph، با معماری انعطاف‌پذیر خود از جریان‌های کنترلی مختلف (مانند عامل‌های تک‌عاملی، چندعاملی، سلسله‌مراتبی و ترتیبی) پشتیبانی کرده و برای سناریوهای پیچیده و واقعی طراحی شده است.

Gemini Flash 1.5 API

مدل Gemini Flash 1.5 از جمله مدل‌های زبانی بزرگ شرکت گوگل است که برای وظایف متنوع زبان‌شناختی طراحی شده است. این مدل نسخه‌ی رایگان محدودی دارد که برای تمرین حاضر کاملاً مناسب است. برای استفاده از این مدل در LangGraph می‌توانید از کلاس ChatGoogleGenerativeAI در ماژول langchain_google_genai استفاده کنید. (در هربار اجرا اگرچه اولین فراخوانی به API ممکن است زمان‌بر باشد، اما سرعت اجرای آن در ادامه مطلوب است.)

LlamaParse

ابزار LlamaParse برای استخراج داده‌های با کیفیت از فایل‌های متنی طراحی شده است و عملکرد بسیار خوبی در استخراج اطلاعات از فرمت‌هایی مانند PDF دارد. این ابزار که به صورت API به عنوان بخشی از پلتفرم LlamaCloud ارائه می‌شود، امکان استخراج داده‌های تمیز و ساختارمندی برای استفاده در فرآیندهای پیشرفته‌ای مانند RAG را به راحتی آماده می‌کند.

LanceDB

LanceDB یک پایگاه‌داده برداری قدرتمند و متن‌باز است که برای ذخیره‌سازی embeddings و جستجو براساس آنها طراحی شده است. این ابزار با کاربری ساده و امکانات پیشرفته، انتخابی ایده‌آل برای ذخیره‌سازی داده‌های مبتنی بر بردار و استفاده در برنامه‌های RAG، جستجوی برداری مقیاس‌پذیر، و حتی کاربرد در داده‌های حجیم اپلیکیشن‌های هوش مصنوعی است.

Chainlit

ابزار Chainlit یک فریم‌ورک متن‌باز پایتونی است که توسعه‌دهندگان را قادر می‌سازد تا در مدت کوتاهی رابط‌های کاربری برای چت‌بات‌ها یا برنامه‌های عاملی بسازند. این ابزار با پشتیبانی از سناریوهای متنوع، امکان ساخت رابط‌کاربری‌ای مشابه ChatGPT و ایجاد تجربه‌های سفارشی برای کاربران را فراهم می‌کند. با استفاده از Chainlit می‌توانید یک تجربه‌ی حرفه‌ای و کارآمد را برای چت‌بات خود در کوتاه‌ترین زمان ممکن ایجاد کنید.

توضیح کد ارتباط با پایگاه داده

دو جدول اصلی برای پایگاه داده وجود دارد:

۱. **جدول سفارش‌ها food_orders:**
این جدول اطلاعات مربوط به سفارش‌ها را شامل می‌شود. ستون‌های این جدول به شرح زیر است:

- id شناسه‌ی سفارش
- person_phone_number شماره تماس فرد
- person_name نام فرد سفارش‌دهنده
- status وضعیت سفارش که یکی از مقادیر preparation، delivery، canceled یا delivered را می‌پذیرد
- order_description توضیحی درباره‌ی محتویات سفارش
- comment نظر ثبت شده مربوط به سفارش

id	person_phone_number	person_name	status	order_description	comment
1	123-456-7890	Alice	preparation	Kebab and Chicken Kebab	
2	444-333-2222	Diana	delivered	Steak	**Not Good!**

۲. **جدول غذاها foods**
این جدول اطلاعات مربوط به غذاها را شامل می‌شود. ستون‌های این جدول عبارتند از:

- id شناسه‌ی غذا
- food_name نام غذا
- food_category دسته‌بندی غذا مانند fast_food، persian_food و غیره
- restaurant_name نام رستوران
- price قیمت غذا

id	food_name	food_category	resturant_name	price
1	Kebab	persian_food	Rumi	**12.99**
2	Pizza	fast_food	PizzaHot	**8.5**

توابع پایگاه داده

در این تمرین، چند تابع اصلی برای ارتباط و مدیریت پایگاه داده نوشته شده است:

۱. **food_search**
این تابع برای جستجوی غذا براساس نام غذا، نام رستوران، یا هر دو استفاده می‌شود. از الگوریتم فاصله ویرایشی (Levenshtein Distance) برای یافتن تطابق نزدیک استفاده می‌کند. نتایج براساس میزان تطابق مرتب شده و بازگردانده می‌شوند.

۲. **cancel_order**
این تابع برای لغو سفارش‌هایی که در وضعیت preparation هستند، استفاده می‌شود. اگر سفارش در وضعیت دیگری باشد یا شناسه‌ی آن وجود نداشته باشد، پیام خطا برمی‌گرداند. برای این تابع شماره سفارش و شماره موبایل باید مربوط به یک سفارش باشند حتما.

۳. **comment_order**
این تابع امکان افزودن یا به‌روزرسانی توضیحات و نظرات مرتبط با یک سفارش را فراهم می‌کند. اگر سفارش با شناسه‌ی داده‌شده وجود نداشته باشد، پیام خطا بازگردانده می‌شود.

۴. **check_order_status**
این تابع وضعیت فعلی یک سفارش را بررسی و برمی‌گرداند. در صورت نبودن سفارش با شناسه‌ی داده‌شده، پیام خطا نمایش داده می‌شود.

مشاهده نمونه استفاده

برای آشنایی بیشتر با نحوه‌ی استفاده از توابع پایگاه داده، می‌توانید فایل **db_manager_sample_use.py** را بررسی کنید. این فایل شامل مثال‌هایی از جستجوی غذا، لغو سفارش، افزودن نظر به سفارش و بررسی وضعیت سفارش است که به درک بهتر نحوه‌ی کار با پایگاه داده کمک می‌کند.

در طراحی چت‌بات، این توابع پایگاه داده مستقیماً به‌عنوان ابزار (tool) یا گره (node) مورد استفاده قرار نمی‌گیرند، بلکه توابع جدیدی طراحی می‌شوند که با استفاده از این توابع پایگاه داده، منطق‌ها، مدیریت خطا و سایر نیازمندی‌های چت‌بات را نیز پوشش می‌دهند.

۱- پاسخ‌دهی به اطلاعات عمومی و تخصصی غذاها (۳۰ نمره)

شرح کلی

در این بخش، چت‌فود به سوالات کاربران درباره‌ی خوراکی‌ها و غذاها پاسخ می‌دهد. هدف این است که کاربر بتواند اطلاعات عمومی و تخصصی در مورد غذاهای مختلف دریافت کند. این پاسخ‌ها باید با استفاده از معماری **Agentic RAG** تولید شوند. ابتدا بررسی می‌شود که آیا سوال کاربر قابل پاسخ‌گویی براساس کتاب مرجع داده‌شده (*The New Complete Book of Food*) است یا خیر. اگر پاسخ در کتاب موجود بود، اطلاعات مربوطه بازیابی شده و به‌عنوان **context** برای مدل زبانی ارسال می‌شود تا پاسخی تولید کند.

در صورتی که پاسخ در کتاب موجود نباشد یا به اطلاعاتی اشاره کند که کتاب شامل آن نمی‌شود، چت‌فود از **Tavily API** برای جستجو در اینترنت استفاده می‌کند. اطلاعات بازیابی‌شده از اینترنت نیز به‌عنوان context برای تولید پاسخ استفاده می‌شود. در هر دو حالت، **فیلتری با کمک LLM** انجام می‌شود تا اطمینان حاصل شود که contextهای نامرتبط به سوال کاربر به مدل ارسال نشود. همچنین اگر سوال کاربر خارج از محدوده‌ی عملکرد چت‌فود باشد، باید محترمانه اعلام شود که امکان پاسخ‌دهی وجود ندارد.

نیازمندی‌ها

- پاسخ کاربر در صورت امکان باید براساس اطلاعات بازیابی‌شده از کتاب مرجع تولید شود.
- اگر پاسخ در کتاب موجود نباشد، اطلاعات مورد نیاز از اینترنت جستجو شده و برای تولید پاسخ استفاده شود.
- قبل از ارسال context به مدل، عملیات فیلترینگ برای حذف contextهای نامرتبط انجام شود.
- سوالات خارج از حوزه‌ی عملکرد چت‌فود باید با پیام مناسبی پاسخ داده شوند.

موارد فنی

- برای پردازش فایل PDF کتاب مرجع و تبدیل آن به متن قابل پردازش از **LlamaParse API** استفاده کنید.
- متن پردازش‌شده باید با استفاده از **RecursiveCharacterTextSplitter** به chunkهای کوچک‌تر تقسیم شود. پارامترهای `chunk_size` و `chunk_overlap` را به گونه‌ای تنظیم کنید که chunkها هم‌پوشانی کافی برای حفظ انسجام داشته باشند.
- با استفاده از مدل **BAAI/bge-small-en-v1.5** از **sentence_transformers** هرکدام از chunkها را به **embeddings** تبدیل کنید و این embeddings را در پایگاه داده‌ی **LanceDB** ذخیره کنید.
- بازیابی اطلاعات از پایگاه داده با روش بازیابی ترکیبی انجام شود.
- برای جستجو در اینترنت از **Tavily API** استفاده کنید و اطلاعات بازیابی‌شده را مشابه context کتاب مرجع، فیلتر کنید.

سوالات

- چرا دادگان را به chunk تبدیل می‌کنیم؟ تاثیر chunk_size و chunk_overlap چیست؟
- چرا بهتر است که بازیابی ترکیبی باشد؟ بازیاب ترکیبی به چه معنی است؟
- چرا قبل از تولید پاسخ، contextهای بازیابی یا جستجو شده را فیلتر می‌کنیم؟
- مدل قابلیت پاسخ به سوالات را دارد. چرا کلاً این وظیفه را به مدل نمی‌سپاریم و RAG پیاده‌سازی کرده‌ایم؟

نمونه سناریو

ورودی کاربر : پاستا چیه؟

عملیات : بانوجه به اینکه این سوال امکان پاسخگویی از متن کتاب مرجع را دارد، ورودی پردازش شده و context مربوط بازیابی می‌شود. سپسcontext های بازیابی‌شده با کمک مدل فیلتر شده و اطلاعات نامرتبط حذف می‌شوند. اطلاعات مرتبط به مدل داده می‌شود و پاسخ نهایی تولید و نمایش داده می‌شود.

ورودی کاربر: قورمه‌سبزی چه غذاییه؟

عملیات : به طور کلی مشابه عملیات سوال قبلی است با این تفاوت که درباره‌ی غذاهای ایرانی در کتاب مرجع، اطلاعاتی نیامده است. پس contextها را از اینترنت جستجو کرده و براساس آنها پاسخ را تولید می‌کند.

ورودی کاربر: چرا آسمان آبی است؟

عملیات :این سوال خارج از محدوده عملکرد چت‌فود است. پس چت‌فود مودبانه باید پاسخ دهد که امکان پاسخ‌دهی به این سوال را ندارد.

۲- خدمات مشتریان (۱۵ نمره)

شرح کلی

در این بخش، چت‌فود به کاربران اجازه می‌دهد عملیات مربوط به سفارش‌های خود را مدیریت کنند. سه قابلیت اصلی این بخش شامل لغو سفارش، پیگیری وضعیت سفارش، و ثبت نظر برای سفارش است. کاربران باید بتوانند از طریق چت‌بات و به زبان طبیعی این عملیات را انجام دهند. چت‌بات با توابع مرتبط پایگاه داده در ارتباط خواهد بود و عملیات خواسته‌شده را اجرا می‌کند یا پیام مناسب را نمایش می‌دهد.

چت‌بات باید به درستی ورودی‌های لازم مانند شماره سفارش و شماره موبایل را از کاربران دریافت کند و اگر کاربر اطلاعات لازم را ناقص یا اشتباه وارد کرد، وضعیت را مدیریت کرده و درخواست ورودی‌های صحیح را تکرار کند. همچنین اگر عملیات موردنظر به دلیل شرایط خاص قابل انجام نباشد (مانند تلاش برای لغو سفارش با وضعیت *delivered*)، چت‌فود باید پیام مناسبی به کاربر نمایش دهد.

نیازمندی‌ها

- امکان انجام عملیات لغو سفارش، پیگیری وضعیت سفارش، و ثبت نظر برای سفارش با ارتباط به توابع پایگاه داده وجود داشته باشد.
- در صورت موفقیت‌آمیز بودن عملیات یا بروز خطا، پیام مناسب به کاربر نمایش داده شود.
- ورودی‌های لازم مانند شماره سفارش و شماره موبایل باید از طریق چت و به زبان طبیعی از کاربر دریافت شود.
- مدیریت شرایط مختلف مانند وارد نکردن اطلاعات، ارائه اطلاعات ناقص یا اشتباه، و تکرار درخواست برای ورودی‌های صحیح به درستی انجام شود.

موارد فنی

- برای لغو سفارش ابزار یا گره‌ای بسازید که از تابع `cancel_order(order_id)` استفاده می‌کند. این تابع بررسی می‌کند. براساس خروجی تابع، باید نمایش پیغام مناسب به کاربر با استفاده از LLM را مدیریت کند. برای دو عملیات دیگر نیز از توابع مربوط به خودشان در ابزار یا گره‌ای که می‌نویسد استفاده کنید.
- در همه موارد، ارتباط با توابع باید به گونه‌ای انجام شود که خطاهای احتمالی (مانند وارد کردن شناسه سفارش نامعتبر) به‌درستی به کاربر بیان شوند.

نمونه سناریو (صرفا مثال است)

لغو سفارش

ورودی کاربر : می‌خوام سفارشم رو کنسل کنم.

چت‌فود : برای لغو سفارش به شماره سفارش و شماره موبایل شما نیاز دارم.

ورودی کاربر : شماره سفارشم ۱۲۳ هست.

چت‌فود: لطفاً شماره موبایل خود را هم وارد کنید.

ورودی کاربر :۱۲۳۴-۵۶۷-۸۹۰.

در این مرحله چت‌فود عملیات مربوط را در پشت زمینه انجام می‌دهد و براساس آن پیغام مناسب را به کاربر نمایش می‌دهد.

پیگیری وضعیت سفارش

ورودی کاربر :می‌خوام بدونم سفارش شماره ۴۵۶ در چه وضعیه؟

چت‌فود در این مرحله در پشت زمینه عملیات مربوط به چک کرده و به طور مثال این سفارش تگ آماده‌سازی دارد.

چت‌فود :سفارش شما در وضعیت آماده‌سازی است.

۳- جستجوی غذا (۱۰ نمره)

شرح کلی

در این بخش، کاربران می‌توانند با استفاده از چت‌فود به‌جای نوار جستجوی اپلیکیشن فرضی سفارش غذا، اطلاعات مورد نظر خود را درباره‌ی غذاهای موجود در رستوران‌ها پیدا کنند. جستجو شامل مواردی مانند نام غذا، نام رستوران، یا ترکیبی از این دو است. چت‌فود باید بتواند با درک ورودی‌های کاربر و ارسال آنها به تابع جستجوی پایگاه داده، نتایج مناسب را ارائه دهد. پاسخ نهایی به‌صورت زبان طبیعی و در قالب گفتگو به کاربر بازگردانده می‌شود. این قابلیت باید تجربه‌ای مشابه با جستجو در یک اپلیکیشن واقعی را برای کاربران فراهم کند.

نیازمندی‌ها

- کاربران بتوانند با زبان طبیعی و در قالب گفتگو، جستجوهای خود را انجام دهند.
- جستجو صرفاً بر اساس نام غذا، نام رستوران، یا ترکیبی از این دو انجام شود.
- چت‌فود باید توانایی درک ورودی‌های مختلف کاربر و ارائه‌ی پاسخ متناسب را داشته باشد.

موارد فنی

- از تابع `food_search(food_name=None, restaurant_name=None)` برای جستجوی غذا در پایگاه داده استفاده کنید.
- ورودی‌های کاربر (مانند نام غذا یا رستوران) باید از متن گفتگو استخراج و به تابع جستجو ارسال شوند.
- نتایج بازگردانده‌شده از تابع باید به مدل ارسال شوند تا پاسخ به زبان طبیعی تولید شده و به کاربر ارائه شود.
- چت‌فود باید درک درستی از روند گفتگو داشته باشد و اگر ورودی‌های ناقص یا گمراه‌کننده‌ای ارائه شد، درخواست اطلاعات تکمیلی کند.

نمونه سناریو ۱

ورودی کاربر: قورمه‌سبزی الان کدوم رستوران‌ها دارن و چنده؟

در پشت زمینه با استفاده از تابع مربوطه، جستجو در پایگاه داده انجام می‌شود و اطلاعاتی مانند این بازگردانده می‌شود.

چت‌فود :قورمه‌سبزی در رستوران میلاد با قیمت ۱۲۵ هزارتومان و در رستوران پارس با قیمت ۱۳۰ هزارتومان موجود است.

ورودی کاربر :پیتزا پپرونی رستوران میلاد و پسران چنده؟

در پشت زمینه، جستجو در پایگاه داده انجام می‌شود و اطلاعات بازگردانده‌شده به مدل داده می‌شود.

چت‌فود : پیتزا پپرونی در رستوران میلاد و پسران با قیمت ۲۲۰ هزارتومان موجود است.

نمونه سناریو ۲ ورودی ناقص یا گمراه‌کننده

ورودی کاربر :پیتزا می‌خوام.

چت‌فود : آیا رستوران خاصی مد نظر شماست؟

ورودی کاربر : نوچ.

در پشت زمینه، جستجو انجام می‌شود و اطلاعاتی مانند رستوران‌ها و قیمت‌های پیتزا نمایش داده می‌شود.

۴- پیشنهاد غذا (۳۵ نمره)

شرح کلی

این بخش به کاربران کمک می‌کند که غذایی را براساس سلیقه و تمایلشان انتخاب کنند، زمانی که دقیقاً نمی‌دانند چه غذایی می‌خواهند. می‌توانید از معماری‌ها و روش‌های عاملی مانند **ReAct**، **Reflexion**، و **Plan and Execute** الهام بگیرید تا فرآیندی چندمرحله‌ای برای پیشنهاد غذا داشته باشید.

پیشنهاد غذا براساس دانش درونی مدل، ورودی کاربر، و بررسی موجودی غذاها انجام می‌شود. در این بخش، چت‌فود باید ابتدا نیاز کاربر را تحلیل کرده، غذاهای مناسب را شناسایی کند و سپس با استفاده از بخش جستجوی غذا بررسی کند که آیا غذاهای پیشنهادی موجود هستند یا خیر. در صورتی که غذایی مطابق تمایل کاربر موجود نباشد، چت‌فود باید پیشنهاد دیگری ارائه دهد.

نیازمندی‌ها

- چت‌فود باید بتواند براساس ورودی کاربر، تحلیل‌های لازم را انجام داده و یک یا چند غذا را به کاربر پیشنهاد دهد.
- اطمینان حاصل شود که غذاهای پیشنهادی در رستوران‌های موجود هستند و با تمایل کاربر (مانند دسته‌بندی و طعم) مطابقت دارند.
- در صورت عدم موجودی غذا یا عدم رضایت کاربر، چت‌فود باید به‌طور هوشمند پیشنهاد دیگری ارائه دهد.

موارد فنی

- از معماری‌های مانند **Reflexion** یا **Plan and Execute** الهام بگیرید تا فرآیند فکر کردن و تحلیل مدل را شبیه‌سازی کنید.
- در این بخش از قابلیت **structured output** استفاده کنید تا خروجی مدل به‌صورت سازمان‌یافته باشد و ارتباط مناسبی با اجزای درونی سیستم برقرار شود.
- پیشنهاد غذا باید براساس اطلاعات بازیابی‌شده از پایگاه داده و موجودی غذاها انجام شود.

نمونه سناریو:

ورودی کاربر :یه غذای فست‌فودی تند هوس کردم. چی پیشنهاد میدی؟

چت فود در پشت زمینه تحلیل می‌کند که کاربر به دنبال غذای فست‌فودی و تند است. غذاهایی مانند «هات‌داگ تند»، «پیتزا پپرونی» و «فیله استریپس اسپایسی» را مورد بررسی قرار می‌دهد. با استفاده از بخش جستجوی غذا، موجودی این غذاها را بررسی می‌کند. براساس نتایج به دست آمده، از مدل می‌خواهد تا پاسخ دهد.

چت‌فود: پیتزا پپرونی در رستوران میلاد و پسران موجود است و قیمت آن ۲۲۰ هزارتومان است.

۵- رابط کاربری (۱۰ نمره)

رابط کاربری چت‌فود باید با استفاده از **Chainlit** پیاده‌سازی شود. این رابط جایگزینی برای ترمینال برنامه‌نویسی است و تعاملات چت‌فود را در یک محیط گرافیکی چت نمایش می‌دهد. توصیه می‌شود ابتدا از طریق ترمینال چت‌فود را طراحی و تکمیل کنید و در آخرین مرحله به رابط کاربری آن را متصل کنید.

نیازی به پیاده‌سازی قابلیت‌های پیشرفته‌تر Chainlit مانند **stream** یا **steps** نیست. رابط کاربری باید ساده باشد و پس از هر بار رفرش صفحه، حافظه‌ی چت پاک شود و از ابتدا اجرا شود. با این حال، در یک جلسه‌ی چت، کاربر باید بتواند به تاریخچه‌ی پیام‌های قبلی خود دسترسی داشته باشد و آن‌ها را دنبال کند.

بخش‌های امتیازی (۱۵ امتیاز)

۱- مدیریت بهینه چت‌های طولانی (۵ امتیاز)

در مواردی که حجم پیام‌ها در یک جلسه چت طولانی می‌شود، نگهداری تمامی پیام‌ها می‌تواند هزینه‌بر باشد و عملکرد چت‌بات را تحت تأثیر قرار دهد. برای مدیریت بهینه چت‌های طولانی، روش‌های مختلفی مانند **trim** (حذف پیام‌های قدیمی، filter (حذف پیام‌های غیرضروری)، و summarize (خلاصه‌سازی پیام‌ها) وجود دارد. در این بخش، شما باید با تحلیل مزایا و معایب هر روش، یکی از این رویکردها را انتخاب کرده و پیاده‌سازی کنید.

۲- نمایش به صورت stream و مراحل در حال انجام (۵ امتیاز)

برای بهبود تجربه کاربری، پیام‌های چت‌فود باید به‌صورت stream نمایش داده شوند. همچنین در فرآیندهایی که شامل مراحل مختلف (مانند جستجو در اینترنت، تحلیل داده‌ها، یا استفاده از ابزارها) هستند، هر مرحله باید به‌طور جداگانه و در زمان اجرای آن نمایش داده شود. این قابلیت باید با استفاده از امکانات Chainlit و **LangGraph** پیاده‌سازی شود و پیاده‌سازی این بخش در ترمینال کافی نیست و باید در رابط کاربری انجام شود.

۳- دریافت تأییدیه و human in the loop (۵ امتیاز)

برای برخی از عملیات‌های حساس، مانند لغو سفارش، باید از کاربر تأییدیه دریافت شود. به عنوان مثال، قبل از لغو سفارش، چت‌فود باید توضیحات سفارش را نمایش دهد و از کاربر تأیید بگیرد که آیا می‌خواهد این سفارش را لغو کند یا خیر. این قابلیت باید با استفاده از human in the loop و امکانات موجود در **LangGraph** پیاده‌سازی شود. پیاده‌سازی دستی این قابلیت پذیرفته نیست و باید از ابزارهای استاندارد برای این کار استفاده شود. همچنین باید اطمینان حاصل شود که این قابلیت باعث اختلال در عملکرد رابط کاربری نشود و تجربه‌ای روان و کاربرپسند ارائه دهد.

ملاحظات (حتما مطالعه شود)

تمامی نتایج شما باید در یک فایل فشرده با عنوان NLP-CA6-StudentID تحویل داده شود.

- خوانایی و دقت بررسی‌ها در گزارش نهایی از اهمیت ویژه‌ای برخوردار است. به تمرین‌هایی که به صورت کاغذی تحویل داده شوند یا به صورت عکس در سایت بارگذاری شوند، ترتیب اثری داده نخواهد شد. دقت کنید که حتما گزارشات خود را در قالب ارائه شده برای تحویل تکالیف که در سامانه برای شما بارگذاری شده است ارسال بفرمایید.
- کدهای نوشته شده برای هر بخش را با نام مناسب مشخص کرده و به همراه گزارش تکلیف ارسال کنید. همه‌ی کدهای پیوست گزارش بایستی قابلیت اجرای مجدد داشته باشند. در صورتی که برای اجرا مجدد آن‌ها نیاز به تنظیمات خاصی می‌باشد بایستی تنظیمات مورد نیاز را نیز در گزارش خود ذکر کنید. **دقت کنید که تمامی کدها باید توسط شما اجرا شده باشند و نتایج اجرا در فایل کدهای ارسالی مشخص باشد. به کدهایی که نتایج اجرای آن‌ها در فایل ارسالی مشخص نباشد نمره‌ای تعلق نمی‌گیرد.**
- این تمرین (تمرین آخر) امکان ارسال با تاخیر ندارد. باتوجه به قوانین درسی که در ابتدای ترم برای شما بارگذاری شد، تمرین آخر شامل ارسال با تاخیر (**با یا بدون گریس**) نمی‌شود.
- **توجه کنید این تمرین باید به صورت تک نفره انجام شود و پاسخ‌های ارئه شده باید نتیجه فعالیت فرد نویسنده باشد (همفکری و به اتفاق هم نوشتن تمرین نیز ممنوع است). در صورت مشاهده تشابه به همه افراد مشارکت کننده، نمره صفر تعلق می‌گیرد و به استاد نیز گزارش می‌گردد.**
- در صورت بروز هرگونه مشکل با ایمیل زیر در ارتباط باشید:

miladmohammadi@ut.ac.ir

تاریخ آپلود تمرین	۱۷ دی ۱۴۰۳
مهلت تحویل بدون جریمه	۳۰ دی ۱۴۰۳
مهلت تحویل با تأخیر، با جریمه ۱۰ درصد	ندارد!