
بسم الله الرحمن الرحيم

عنوان:

ChatFood: A Multi-Agent Conversational AI for Food Ordering

(چت‌فود: یک هوش مصنوعی محاوره‌ای چند-ایجتی برای سفارش غذا)

نگارنده:

مجید خرمگاه

مهر 1404

چکیده

پروژهChatFood ، پیاده‌سازی یک دستیار هوشمند و مکالمه‌ای برای یک اپلیکیشن فرضی سفارش غذاست. این سیستم با بهره‌گیری از چارچوب **LangGraph**، بر پایه یک **معماری چند-ایجتی (Multi-Agent)** طراحی شده است که در آن، یک مسیریاب هوشمند وظایف را به صورت دینامیک به ایجنت‌های متخصص واگذار می‌کند. پروژه با موفقیت تمام اهداف اولیه تسک، شامل پیاده‌سازیRAG ، مدیریت سفارش و جستجوی غذا را محقق کرده و با افزودن قابلیت‌های پیشرفته‌ای نظیر **پیشنهاد فعال و شخصی‌سازی شده، حافظه مکالمه پایدار، رابط کاربری کاملاً تعاملی با سبد خرید،** و **بهینه‌سازی عملکرد،** از یک تسک آزمایشی به یک نمونه اولیه محصول (Prototype) حرفه‌ای و قابل اتکا تبدیل شده است.

فهرست مطالب

مقدمه 1

اهداف اولیه پروژه 1

فصل اول: معماری سیستم 2

1-1. فلسفه طراحی 3

1-2. نمودار جریان کاری 4

فصل دوم: معرفی اجزا(Components Deep Dive) 5

2-1. مسیریاب هوشمند(Router) 6

2-2. ایجنت پیشنهاددهنده فعال (Recommendation Agent) 6

2-3. ایجنت جستجوی فیلتردار(FilterAgent) 6

2-4. ایجنت جستجوی ساده (FoodSearchAgent) 6

2-5. ایجنت مدیریت سبد خرید (CartAgent) 7

2-6. ایجنت مدیریت سفارش (OrderManager) 7

2-7. ایجنت اطلاعات (InformationAgent - RAG) 7

فصل سوم: ویژگی‌های کلیدی و نوآوری‌ها 8

3-1. حافظه مکالمه پایدار 9

3-2. تعامل فعال(Proactive Engagement) 9

3-3. معماری مستحکم UI با کانال داده مجزا 9

3-4. بهینه‌سازی عملکرد(Caching) 9

3-5. مدیریت خطای جامع 9

فصل چهارم: پشته فناوری(Technology Stack) 10

فصل پنجم: راهنمای نصب و اجرا 12

فصل ششم..... 14

چالش‌های فنی و راه‌حل‌ها..... 14

6-1. مشکلات..... 15

6-1-1. چالش..... 15

6-1-2. مشاهده..... 15

6-1-3. راه حل..... 15

6-2-1 . چالش..... 15

6-2-2 . مشاهده..... 15

6-2-3 . راه حل..... 15

6-3-1. چالش..... 16

6-3-2. مشاهده..... 16

6-3-3. راه حل..... 16

فصل هفتم..... 17

مسیر توسعه آینده(Future Roadmap)..... 17

مقدمه

در اکوسیستم رقابتی اپلیکیشن‌های سفارش غذا، ارائه یک تجربه کاربری هوشمند، سریع و شخصی‌سازی شده یک مزیت کلیدی است. چت‌بات‌های سنتی اغلب در درک مکالمات چندمرحله‌ای و ارائه تعاملات پویا ناتوان هستند. هدف پروژه ChatFood ، ساخت یک دستیار مجازی است که این خلاء را با استفاده از آخرین تکنیک‌های هوش مصنوعی محاوره‌ای پر کند.

اهداف اولیه پروژه

این پروژه با اهداف زیر که توسط شرکت مهیمن تعریف شده بود، آغاز گردید:

- آشنایی با **LangGraph** برای ساخت سیستم‌های مبتنی بر ایجنت.
- یکپارچه‌سازی ابزارهایی مانند LanceDB (برایRAG) و Chainlit (برایUI).
- پیاده‌سازی معماری‌های **RAG** و **ReAct**.
- پوشش دادن ویژگی‌های اصلی شامل جستجوی غذا، مدیریت سفارش و پرسش و پاسخ.

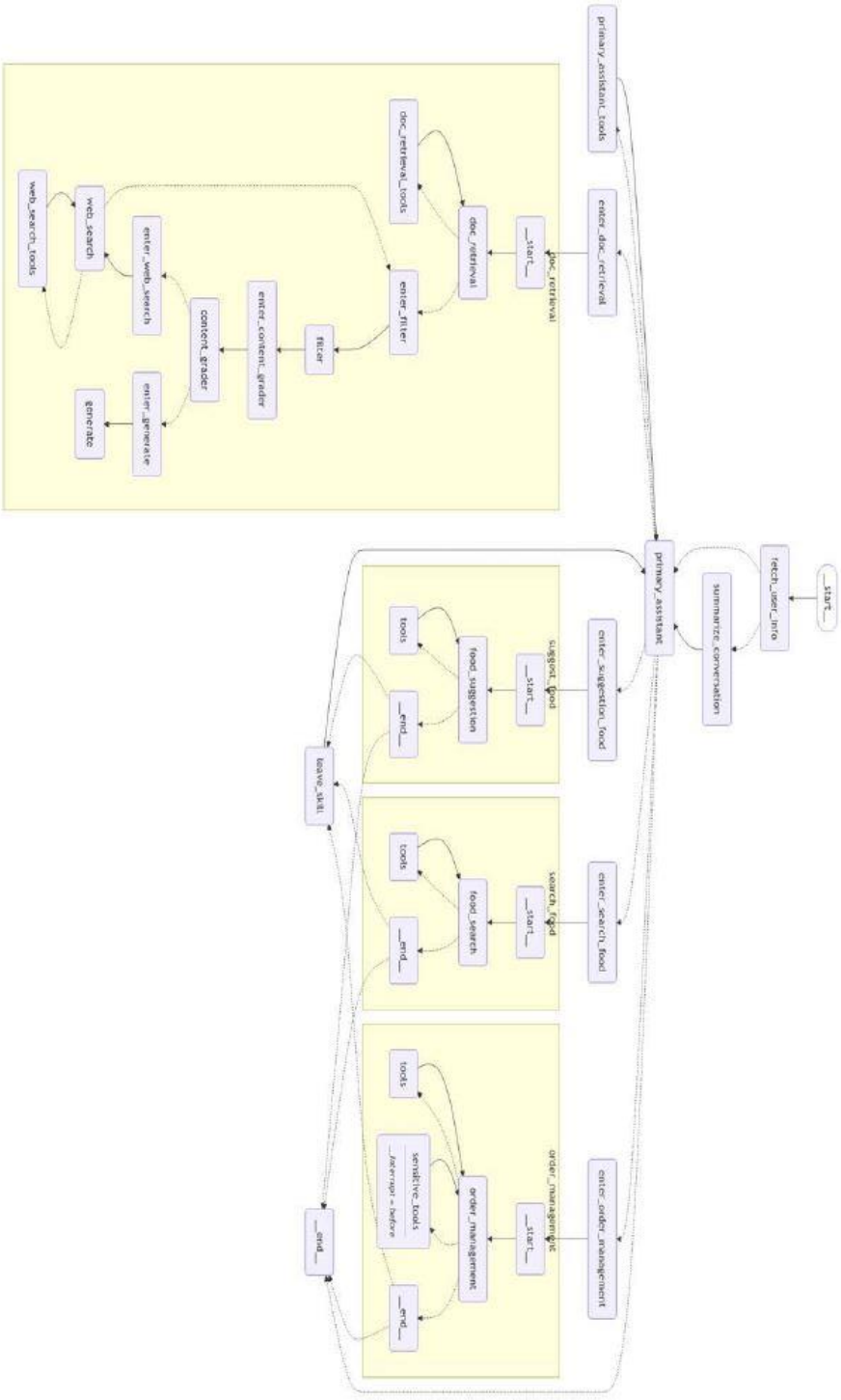
فصل اول

معماری سیستم

1-1. فلسفه طراحی

معماری ChatFood بر پایه الگوی **"Router/Dispatcher"** در یک سیستم چند-ایجتی بنا شده است. به جای یک مدل زبان بزرگ یکپارچه که تمام وظایف را انجام دهد، ما از مجموعه‌ای از **ایجنت‌های کوچک و متخصص** استفاده می‌کنیم که هر کدام برای یک کار خاص (مانند جستجو، مدیریت سفارش) بهینه شده‌اند. این رویکرد ماژولار، قابلیت نگهداری، توسعه و دیباگ کردن سیستم را به شدت افزایش می‌دهد. **LangGraph** به عنوان ارکستراتور اصلی، وضعیت (State) گفتگو را مدیریت کرده و جریان کار را بین این ایجنت‌ها هدایت می‌کند.

1-2. نمودار جریان کاری



فصل دوم

معرفی اجزا (Components Deep Dive)

2-1. مسیریاب هوشمند(Router)

نقش : مغز متفکر سیستم. این ایجنت اولین گره در گراف اصلی است و با تحلیل آخرین پیام کاربر و کل تاریخچه گفتگو، تصمیم می‌گیرد که کدام ایجنت متخصص باید درخواست را کند.

تکنولوژی : از LLM با قابلیت **Structured Output** برای تولید یک تصمیم قطعی (`'FoodSearch', 'CartAgent', ...`) استفاده می‌کند.

2-2. ایجنت پیشنهاددهنده فعال (Recommendation Agent)

نقش :آغازگر مکالمه. این ایجنت به صورت **فعال (Proactive)** در ابتدای گفتگو، با تحلیل تاریخچه سفارش‌های کاربر از دیتابیسSQLite ، یک پیشنهاد ویژه و شخصی‌سازی شده ارائه می‌دهد.

تکنولوژی async : برای اجرای غیرمسدودکننده در `Chainlit`، پرامپت مهندسی شده برای تولید متن خلاقانه.

2-3. ایجنت جستجوی فیلتردار(FilterAgent)

نقش :پاسخ به سوالات پیچیده و چند شرطی. این ایجنت قادر به درک درخواست‌هایی است که شامل فیلتر بر اساس قیمت، دسته‌بندی و نام غذا به صورت همزمان هستند.

ابزار :ازابزار `advanced_food_search_tool`استفاده می‌کند که منطق فیلتر کردن را در سطح پایگاه داده (SQL) پیاده‌سازی می‌کند تا بهینه‌تر باشد.

2-4. ایجنت جستجوی ساده (FoodSearchAgent)

نقش :پاسخ به سوالات ساده و تک‌مرحله‌ای در مورد منو. (مثال: "پیتزا چی دارید؟").

ابزار:`simple_food_search_tool`.

2-5. ایجنت مدیریت سبد خرید (CartAgent)

نقش: تشخیص درخواست کاربر برای مشاهده سبد خرید. این ایجنت فقط یک سیگنال تولید می‌کند و منطق اصلی مدیریت سبد خرید در لایه UI (Chainlit) پیاده‌سازی شده است.

ابزار: `view_cart_tool`.

2-6. ایجنت مدیریت سفارش (OrderManager)

نقش: تعامل با جدول `orders` در دیتابیس برای بررسی وضعیت یا لغو سفارش.

ابزار: `get_order_status_tool`, `cancel_order_tool`.

2-7. ایجنت اطلاعات (InformationAgent - RAG)

نقش: پاسخ به سوالات عمومی و دانشنامه‌ای

معماری: یک ایجنت ReAct ساده که از دو ابزار استفاده می‌کند:

knowledge_base_retriever_tool (برای جستجو در LanceDB)

و web_search_tool (برای جستجو در اینترنت).

فصل سوم

ویژگی‌های کلیدی و نوآوری‌ها

3-1. حافظه مکالمه پایدار

با مدیریت لیست پیام‌ها در AgentState و cl.user_session، سیستم قادر است زمینه گفتگوهای طولانی را به طور کامل حفظ کند.

3-2. تعامل فعال (Proactive Engagement)

بر خلاف چت‌بات‌های واکنشی، ChatFood می‌تواند گفتگو را با پیشنهادات مرتبط آغاز کند و تجربه کاربری پویاتری ایجاد کند.

3-3. معماری مستحکم UI با کانال داده مجزا

با اضافه کردن فیلد tool_output به AgentState، ما یک کانال امن برای انتقال داده‌های ساختاریافته از ابزارها به رابط کاربری ایجاد کردیم. این معماری از "توهم LLM "جلوگیری کرده و تضمین می‌کند که کارت‌های تعاملی همیشه به درستی و با داده‌های صحیح نمایش داده شوند.

3-4. بهینه‌سازی عملکرد (Caching)

با پیاده‌سازی الگوی Singleton برای بارگذاری مدلEmbedding ، زمان پاسخ‌دهی در اولین جستجوی RAG از چندین دقیقه به کمتر از ۳ ثانیه کاهش یافت که برای یک محصول واقعی حیاتی است.

3-5. مدیریت خطای جامع

با استفاده از بلوک‌های try...exceptدر تمام لایه‌ها (ابزارها، ایجنت‌ها وUI) ، سیستم در برابر خطاهای پیش‌بینی نشده(مانند مشکلات شبکه یا API) مقاوم شده و از کرش کردن جلوگیری می‌کند.

فصل چهارم

فناوری (Technology Stack)

زبان برنامه‌نویسی: Python 3.11

چارچوب هوش مصنوعی: LangChain, LangGraph

مدل زبان بزرگ (LLM): OpenAI GPT-4o-mini

پایگاه داده:

SQLite (برای داده‌های رابطه‌ای: منو، سفارش‌ها)

LanceDB (برای جستجوی وکتوری و RAG)

رابط کاربری: Chainlit

ابزارهای کلیدی دیگر: HuggingFace Sentence Transformers, Pydantic

فصل پنجم

راهنمای نصب و اجرا

```
#Clone the repository and navigate into the directory
git clone https://github.com/makhresearch/ChatFood-AI-Assistant
cd ChatFood-Project

#Create and activate a virtual environment
 python -m venv venv
#On Windows: venv\Scripts\activate
#On macOS/Linux: source venv/bin/activate

#Install all dependencies
pip install -r requirements.txt

#Set up the OpenAI API Key
#Create a .env file and add your key: OPENAI_API_KEY="sk"…-

#Initialize the databases and knowledge base
python setup_database.py
python update_database.py
python setup_rag.py

#Run the application
chainlit run app.py -w
```

فصل ششم

چالش‌های فنی و راه‌حل‌ها

6-1. مشکلات

6-1-1. چالش

پرهزینه بودن و کندی معماری Plan-and-Execute

6-1-2. مشاهده

ایجنت PlannerAgent برای سوالات نسبتاً ساده، چندین فراخوانی متوالی به API OpenAIانجام می‌داد که منجر به برخورد با محدودیت‌های طرح رایگان (Rate Limit Error) و کندی شدید می‌شد.

6-1-3. راه حل

معماری به طور کامل بازطراحی شد. به جای آن، یک ابزار ترکیبی (search_and_filter_food) و یک ایجنت ReAct ساده (FilterAgent) ایجاد شد. این کار منطق پیچیده را به کد پایتون منتقل کرد، تعداد فراخوانی‌های API را بیش از ٪۵۰ کاهش داد و مشکل را به طور کامل حل کرد.

6-2-1 . چالش

کندی شدید در اولین اجرایRAG

6-2-2 . مشاهده

اولین پاسخ از InformationAgent بیش از ۵ دقیقه طول می‌کشید. دیباگ نشان داد که مدل سنگین Embedding در هر بار فراخوانی از نو بارگذاری می‌شود.

6-2-3 . راه حل

با پیاده‌سازی یک کش ساده (Singleton Pattern) در سطح ماژول، اطمینان حاصل شد که مدل فقط یک بار در ابتدای شروع برنامه بارگذاری می‌شود. این بهینه‌سازی، زمان پاسخ را به چند ثانیه کاهش داد.

6-3-1. چالش

ناپایداری در نمایش کارت‌های تعاملی

6-3-2. مشاهده

در ابتدا، نمایش کارت‌ها و دکمه‌ها در UI به صورت تصادفی با شکست مواجه می‌شد. دلیل آن، "خلاقیت LLM "در بازنویسی خروجی ابزارها و تبدیل آن به متن محاوره‌ای بود که ساختار داده را از بین می‌برد.

6-3-3. راه حل

معماری با اضافه کردن یک کانال داده مجزا (tool_output) در AgentState بازطراحی شد. این کار تضمین می‌کند که داده‌های ساختاریافته به صورت دست‌نخورده از ابزار به UI منتقل شوند و نمایش کارت‌ها ۱۰۰٪ قابل اعتماد باشد.

فصل هفتم

مسیر توسعه آینده(Future Roadmap)

- **سیستم کامل احراز هویت و پروفایل کاربری.**
- **تکمیل چرخه سفارش** با قابلیت پرداخت آنلاین و پیگیری زنده.
- **پیشنهادهای هوشمندتر** بر اساس آیتم‌های مکمل (مثلاً پیشنهاد نوشابه با پیتزا).
- **استقرار (Deployment)** روی یک سرور ابری برای دسترسی عمومی.
- **یکپارچه‌سازی با پلتفرم‌های دیگر** مانند واتس‌اپ و تلگرام.