COMP4332/RMBI4310

Big Data Mining and Management
Advanced Data Mining for Risk Management and Business Intelligence (2025 Spring)

Tutorial 9: a Simple LLM Agent-based Conversational Recommender System

TA: Chunyang LI (cliei@connect.ust.hk)

Background: LLM Agents

What is an LLM Agent?

- A system uses a Large Language Model to autonomously execute complex tasks.

Why we use LLM agents?

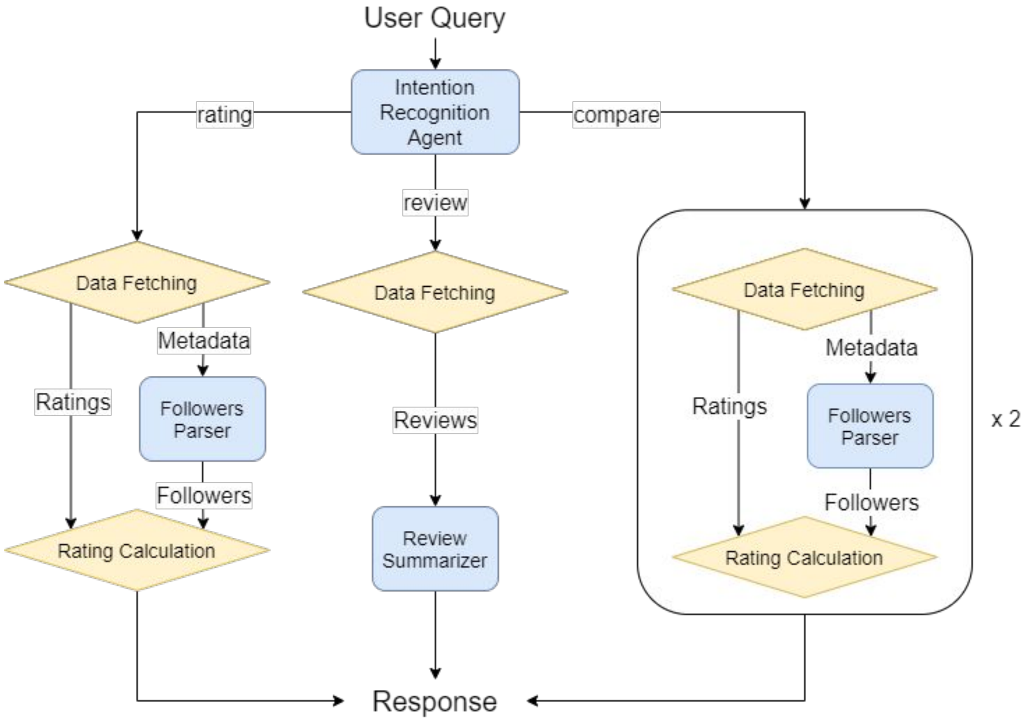
- Understand and generate human-like text.
- Handle complex tasks with minimal hand-coded rules.

In the context of a CRS, LLM agents enable natural language interaction.

Main Workflow

- Use IntentionRecognitionAgent to identify the query type and arguments
- Fetch data from our dataset
- Process the query
  - For “rating”: Compute the rating
  - For “compare”: Compute ratings for two restaurants and compare them
  - For “review”: Summarize reviews using ReviewSummaryAgent
- Returns a user-friendly response

Main Workflow



Different type of user query:

- Tell me something about The Tilt Bar Republic.
- How is KFC rated?
- Hyderabad Hotel and Paradise, which is better?

Tips

What is a well-crafted prompt for LLM in this task:

- Clarity: use clear, unambiguous language to describe the task.
- Output Specification: explicitly state the desired output format and structure.
- Constraints: include constraints to avoid irrelevant or incorrect responses.
- Examples: provide examples of inputs and expected outputs to guide the LLM’s behavior.

Example: sentiment analysis

Task Description:

You want to use an LLM to analyze the sentiment of customer reviews for a product. The LLM should:

- Classify the sentiment of each review as “positive”, “negative”, or “neutral”
- Provide a brief explanation for the classification.
- Output the results in a JSON format for easy processing.

Example: sentiment analysis

Prompt:

You are a sentiment analysis agent for product reviews. Your task is to analyze a list of customer reviews for a product and classify the sentiment of each review as one of the following: "positive," "negative," or "neutral." For each review, provide a brief explanation (1-2 sentences) justifying the sentiment classification.

Return a JSON string where each element is an object with two keys:

- "sentiment": The sentiment label ("positive", "negative", or "neutral").
- "explanation": A brief explanation for the sentiment classification.

Examples:

Input: This phone is fantastic!

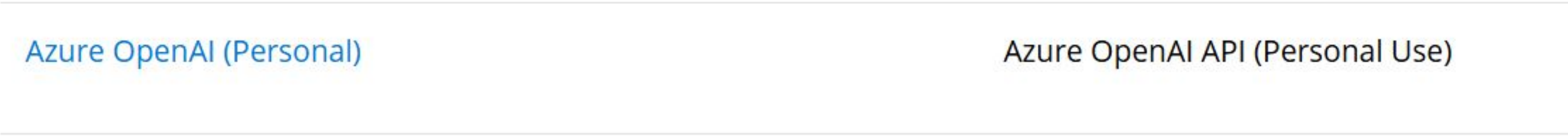
Output: {“sentiment”: “positive”, “explanation”: "The review expresses strong enthusiasm for the phone, indicating a positive opinion."}

Input reviews: {reviews}

Output only the JSON string, with on additional text.

How to generate your azure api key

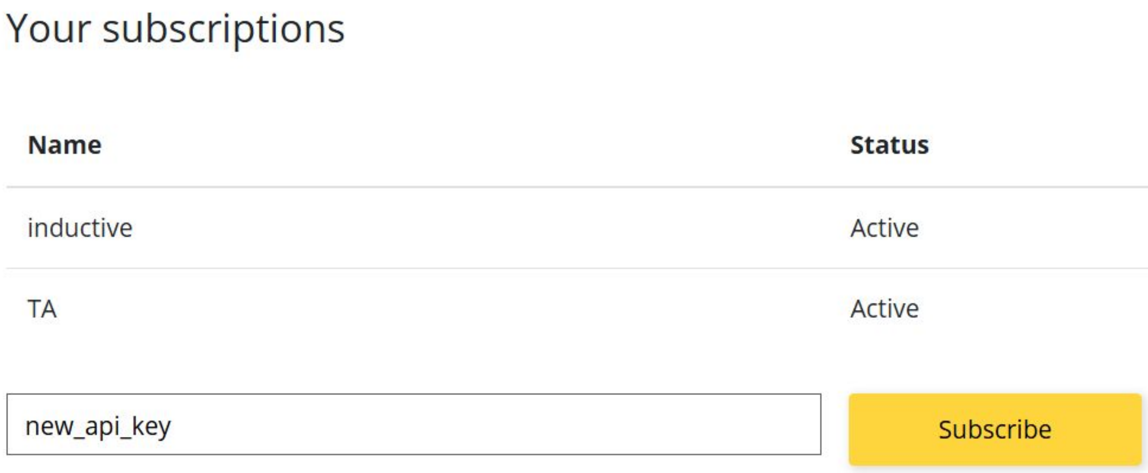
- Navigate to HKUST API Developer Portal Microsoft Azure API Management
- Sign up/Sign in with HKUST account
- Navigate to the Products pape from the menu (top right side)
- Then click on the Azure OpenAI (personal)



Azure OpenAI (Personal)	Azure OpenAI API (Personal Use)

How to generate your azure api key

- Input a description, click on the **Subscribe** button
- Click Show to display the hidden API key. Both Primary and Secondary keys will access the OpenAI API



Your subscriptions

Name	Status
inductive	Active
TA	Active

new_api_key

Thanks