

Project Roadmap: Agentic LLM

Prof. J. Panchal, Arup Chauhan, Swastik Patro, Vinay Prabahkar

Contents

1	Project Brief	2
1.1	Problem Statement	2
1.2	Features	2
1.2.1	Current	2
1.2.2	Future Use Case	2
2	App Structure	2
2.1	Planner & Controller (LLM Agent)	2
2.2	Pipelines	2
3	Technical Specifications	3
3.1	Pretrained LLM	3
3.2	Data Sources	3
3.3	Fine-Tuning	3
4	Current End-to-End Implementation	3
4.1	Query Processing	3
4.2	Data / Information Retrieval for response	3
4.3	Formulating final response	3
5	References	4

1 Project Brief

1.1 Problem Statement

Develop a Siri/Google Assistant-like chatbot that serves as a tour and food guide for Chicago.

1.2 Features

1.2.1 Current

Food / Travel Assistant

- **Image to Text Suggestion**
 - Example: *"Where can I find more dishes like this {image} within 4 miles of me?"*

1.2.2 Future Use Case

- **Itinerary Planning Assistant**
 - **Formal**
 - * Example: *"I have a 3-day holiday block this weekend. I'm interested in visiting museums and attending music concerts. Suggest me 3 different itineraries which I can do within \$300."*
 - **Impromptu**
 - * Example: *"I'm at this place {image}. I have 4 hours of time. What are the things that I can do nearby within walking distance (1 mile)?"*
-

2 App Structure

2.1 Planner & Controller (LLM Agent)

- Breaks down the prompt into subtasks and constructs the respective Directed Acyclic Graph (DAG).
- Resolves missing data by prompting the user for more information.
- Executes the subtasks.
- Outputs the answer to the user.

2.2 Pipelines

- **APIs**
 - Google Distance Matrix
 - Eventbrite
 - Google Places
 - Weather
 - DoorDash/similar
 - Image to Text (to be determined)
 - **RAG (Retrieval-Augmented Generation)**
 - General store of tourist and food data for Chicago.
 - Exploring the possibility of reading Google Q&A or FAQ sections for each place via API call, search call, or RAG.
-

3 Technical Specifications

3.1 Pretrained LLM

Utilize an open-source LLM running locally for processing and generating responses.

3.2 Data Sources

- Web search
- Product data from travel websites for food (itinerary) and places to visit.
- Real-time updates on travel itineraries.
- Grammarly-like tool for advising itineraries for travel agents or end-users.
- Group planning, e.g., editing the itinerary for vegetarian food only.

3.3 Fine-Tuning

- Train the model to create itineraries.
 - Alternatively, create a data structure for itineraries:
 - Places
 - Characteristics
 - Distances
 - Timings
 - Weather
 - Use prompts (agents) or fine-tune on this dataset.
-

4 Current End-to-End Implementation

4.1 Query Processing

- Multimodal Input (Text and Image)
- Identification of information required for response

4.2 Data / Information Retrieval for response

- Approach 1: Direct query from Database
- Approach 2: Using API routes for fetching results

4.3 Formulating final response

- Verifying the retrieved information is relevant and complete.
 - Generating a response in Natural Language
-

5 References

References

- [1] Madaan, Aman, et al. "SELF-REFINE: Iterative Refinement with Self-Feedback."
 - [2] Shinn, Nicholas, et al. "Reflexion: Language Agents with Verbal Reinforcement Learning."
 - [3] Shen, Sheng, et al. "HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face."
 - [4] Wei, Jason, et al. "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models."
 - [5] Wu, You, et al. "AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation."
 - [6] Qian, Zihao, et al. "Communicative Agents for Software Development."
 - [7] Kim, et al. "LLMCompiler: An Agent Compiler for Parallel Multi-Function Planning and Execution."
-
-