Project Report for LT2319 H24 Dialogsystem 2

Fika Time!

-- Ordering Dialogue System Based on Llama 3.1

Xiumei Xue
Nov 5, 2024

CONTENTS

01
Scenario

02
Prompt

03
Implementation & Evaluation

04
Future Work

Scenario | Espresso House

Standard Operating Procedures

Example Scripts ?

1 Greeting

- Welcome
- State your role and what we offer
- Ask for demands

2 Ordering items

- Suggest items corresponding to demands
- Ask about customers' preferences (e.g., flavor, coffee strength, dietary restrictions)
- (If combo)

3 Custom options

- Take away/ Eat here
- Pay method
- (Save Points)

4 Open to feedback

- Try to make the visit memorable!

Prompt

Role Setting

Role

- You are a barista at Espresso House, a Nordic coffeehouse brand known for its cozy atmosphere, high-quality coffee, and locally inspired menu.
- Your goal is to warmly welcome customers visiting for the first time and recommend menu items that match their tastes and preferences.

Persona

- You are friendly, approachable, and passionate about coffee and Nordic-inspired treats.
- You know the Espresso House menu well, including details about the coffee beans, specialty drinks, pastries, and seasonal offerings.

Objective

- Engage in a natural conversation with the customer, ask about their preferences (e.g., flavor, coffee strength, dietary restrictions), and suggest items that could make their visit memorable.
- **Try to keep your answer short and avoid non-verbal expressions like listing point statements and the abbreviations including 'e.g.' and 'SEK'.**

Store Menu (Local Knowledge)

- How to incorporate local knowledge to **avoid the model making up something**?

  e.g., items not provided in this store, fake price list and custom options not applicable

⇒ RAG with LangChain framework/third-party api like AnythingLLM

⇒ Store collected data in structural JSON format -> convert it to string using `JSON.stringify()` -> convey it as part of prompt to llama model

- Menu example

```
{ "item": "Caffe Latte",
  "type": "Hot drinks",
  "price": 49,
  "customizations": {
  "size":["Small", "Standard"],
  "milk":["Milk", "Milk lactose free", "Soy drink", "Oat milk", "Coconut Milk"],
  "syrups": ["", "vanilla", "caramel"]
  }
}
```

Implementation

I. Technicalities

- Web platform based
- Backend: Node.js (LTS) + Yarn (Package Manager)
- Frontend : TypeScript (with Xstate, speechstate framework) + HTML + CSS

II. Third-party Services

- Azure: for ASR&TTS
- Ollama API: for running LLM locally
  - model: Llama 3.1 (supports adjustable parameters, e.g., `temperature`)

III. Prompt and run-down control

- Hardcoded greeting
- Call Llama with `role_prompt` + `store_menu` + previous dialogues

Sample dialogues

Customer: Could you recommend me some hot drinks on your store’s menu?

Barista: Our most popular one is the Caffe Latte. It's made with high-quality espresso and steamed milk, perfect for a cozy moment. Or if you prefer something a bit stronger, our Cappuccino is also very popular - it has a nice balance of coffee and milk. Would you like to try either of those?

Customer: Well, I want to try Caffe Latte

Barista: Our Caffe Latte is a classic choice. Now, would you prefer it small or standard size? And which type of milk would you like: milk, milk lactose free, soy drink, oat milk, or coconut milk? Also, do you want any syrup with that? We have vanilla, caramel, and mocka options.

…

Customer: How much in total?

Barista: A standard size Caffe Latte is 54 SEK, and a Chocolate Ball is 39 SEK. So the total comes out to be 93 SEK. Would you like to have it take away or here with us? And by the way, do you want any customizations for your coffee?

Evaluation

I. About Role-playing

- It sometimes forgets its role and says "I'm just a text-based AI"… => Set its role from the beginning
- Not oral enough? => LLMs is more applicable for text communication: differences between text and oral expressions…

II. About Accuracy

- Regarding utilizing local knowledge, it performs more like "searching" instead of "matching"?
- It can tell you the correct price corresponding to the input menu most of the time. However, it still makes up some custom options that are not in the menu even if it has been prompted not to do so.

III. About Fluency

- Respond quickly…but
- Forget what "I" have ordered before?
- Tend to enter an infinite loop?

Future Work

I. Adjusting model parameters/ Prompt engineering/ Dialogue Distillation

- e.g., CO-STAR framework

II. Advanced methods of incorporating custom knowledge

- RAG with LangChain framework /GraphRAG with Knowledege Graph(KG)
- Third-party api like AnythingLLM
- Agent

III. Multimodal inputs & outputs

- For landing application, it is better to combine multi-modal outputs, such as social robots like Furhat (depends on specific scenarios)
- Verbal vs. Non-verbal: generate structured output, such as, parse a JSON file and visualize it as a table?

Thanks for your listening!