**Interactive Swedish Learning Game — ”Restaurant Ordering” Scenario**

LT2319 H24 Dialogue System2

Final Project

Tianyi Geng

**1.Introduction**

My dialogue system 2 course project extends the language learning game project that I implemented in my dialogue system 1 course project (see: https://github.com/ariageng/Frog\_Game\_v.1.0), which is a language game that combines speech recognition, dialogue management, and natural language understanding. This time, I focused on developing the ”restaurant ordering” scenario, featuring the player/”frog” meeting a cashier at a burger restaurant. Basic Swedish conversations in the corresponding scenario are introduced and the player is encouraged to organize his/her own utterance while suggested sentences are also provided. The project handles the player’s intents by analyzing them through a large language model (”LLaMA3”) and managing dialogue states with the retrieved output of the model. The implementation also attempted for a combination of multimodality (e.g. visual interface elements, recognized texts, synchronized audio and visuals, etc.), which largely enhances natural and engaging user experience.

**2.Technical Components**

The system uses the following services, frameworks and APIs:

* Azure Speech Service (speech recognition, speech synthesis, multilingual voices)
* Xstate (for state update managing), Speechstate (integrates Xstate and Azure APIs)
* LLaMA API (for identifying user intents and generate natural dialogue responses)

User interface:

* Click to talk
* Cute Character GIFs representing the player and the NPC (”Catshier”)
* A few animation changes on the NPC depending on different intents of the player speech
* Speech suggestions emerge on the screen as clickable buttons and play pronunciation audio when clicked
* Order Confirmation UI for order amount adjustment

**3.Data Collection**

The scenario starts with the player/”frog” meeting a cat character(”Catshier”) working at a virtual burger restaurant (”Wax Burger”). The player then gets to talk to Catshier and ask about food options, menu, or recommendations.

The menu has fixed items (food: ”Wax hamburgare”, ”Wax pommes frites”; drinks: ”Wax cola”, ”Wax milkshake”) and the player is only considered making valid orders when at least one item is ordered.

**4.Implementation**

The implementation uses XState for state managing, splitting the interaction according to the communication phases (”Meeting & Introduction”, ”User Input & Intent Recognition”, ”Character Response”, ”Order Confirmation”, ”User Input & Attitude Detection”, ”Order Success/Order Failed”, ”Ending & Character Comments”). The major intents include ”ordering”, ”asking for menu”, ”asking for recommendations”, and ”irrelevant”. Intents other than ”ordering” will trigger the system to go back to listening again. During ordering, several utterance suggestion buttons show up for the player to click and listen to the pronunciation. After an order is made, the NPC will confirm and show the confirmation UI window which allows the player to modify the amount of the items they ordered. After the NPC confirming the order one more time, the system will go to the end scene or go back to the start depending on the user attitude input (using Azure NLU).

**5.Sample Dialogues**

Catshier: ”Hej, vad kan jag göra för dig?”

Frog: ”Hej, vad är meny?”

Catshier: ”Javisst. Vår meny är: Wax hamburgare, Wax pommes frites, Wax cola, Wax milkshake.”

Frog: ”Jag vill ha en hamburgare och en cola.”

Catshier: ”Du beställde en hamburgare och en cola. Något annat?”

[Show order confirmation UI, waiting for the player to adjust the amount of the ordered items]

Frog: ”vad är rekommendation?”

Catshier [smiles]: ”Jag rekommenderar vår specialitet - Wax hamburgare med pommes frites och en milkshake. Det är vår mest populära kombination!”

Frog: ”Jag vill ha en helicopter.”

Catshier [confused look]: ”Jag förstår inte.”

**6.Discussion & Future Plans**

As for results, during implementation tests, the system showed brilliant capability in capturing user intents in most of the cases, while the LLM occasionally giving unexpected responses (e.g. confusing certain intents when speech input is casual, taking ”undefined” as a name of food, etc.).

I found that NLG using LLaMA3 takes longer than natural time and I would attribute that to the complicated prompt I sent to the model to process. Due to the processing time problem, I would attempt to try a combination of LLaMA3 and Azure service for the natural language understanding in the future development of this project (e.g. LLaMA3 for the complicated task that not only identify intents but also return the order data, if exists, to the system in required format; Azure NLU for the simpler task like identifying user attitudes).

For future development, I intend to improve the intent recognition and natural feedback generation (to distinguish better between ”items that can be ordered” and ”items that should not be ordered”).

I also plan to further explore LLM applications in combination with ISU and multimodality (for instance, using LLM to manage and update states directly).