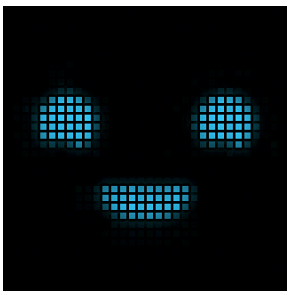
Assignment: Improving Turn-Taking in Human–Robot Interactions

Background



You will be working with a screen-based robot face that can detect faces, listen to speech, and respond with spoken output. The robot works, but its conversation skills are slow and sometimes awkward. Your task is to improve the flow of conversation by combining audio and visual cues, adjusting how the robot manages turn-taking, and testing your improvements with human participants.

Goals

By completing this assignment, you will:

- Explore how audio and camera sensors can be combined for more natural interactions.
- Improve the robot's ability to initiate and manage conversations.
- Implement faster, more human-like turn-taking.
- Evaluate your improvements through a small user study.

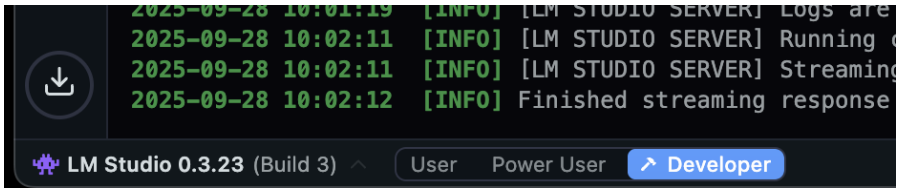
Getting the robot face up and running

To get the robot face up and running, start by downloading and extracting the provided files from Learnit. Once extracted, open a terminal and 'cd' to the project folder. Here you can install the required Python packages by running "**`pip install -r requirements.txt`**" (some systems may require running it as **`pip3`** instead of `pip`). After installation is complete, you can launch the robot by running "**`python main.py`**" (or "**`python3 main.py`**" on systems where Python 3 must be specified). This will start the robot face and allow you to interact with it. One some systems you may have to allow the program access to the microphone and camera for it to work.

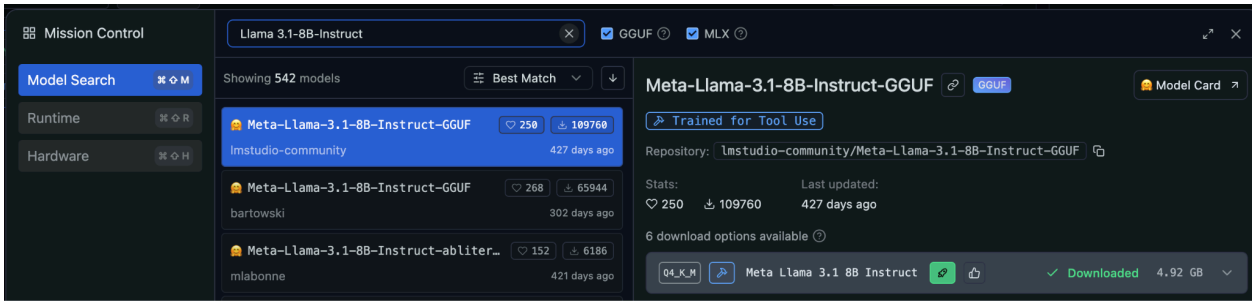
Installing a local Large Language Model (LLM)

The robot face uses a large language model to generate conversation answers. You need to install a program that can make the LLM available for the robot face. To install and run a local large language model, go to https://lmstudio.ai and download LM Studio for your computer. After

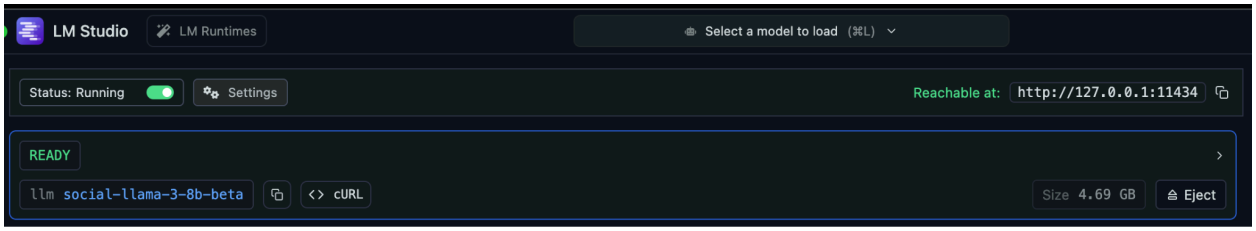
installing, open LM Studio and click the "**Developer**" button at the bottom of the window to enable developer functionality in the program.



Then click the magnifying glass icon on the left to search for models. Search for **`meta-llama-3.1-8b-instruct`** and click Download on the right side.



Once the model is downloaded, click the green Developer icon on the left to open the developer panel. At the top, select your downloaded model from the drop-down menu, then toggle the Status switch to On to enable the model as a server.



Click the Settings button next to the toggle and verify that the server port is set to 11434. After this, the model will be running locally and accessible from the robot face application.

Optional: You can click on the yellow chat button on the left to try chatting with the LLM and see how fast/slow it responds.

Some Implementation Details

Preparation: Explore the code and available sensors

Begin by examining the state variables in **`main.py`**. These give you insight into what the robot can sense and use throughout the interaction.

From the camera (facial tracking data):

- Last_speaking
- last_face_visible
- mouth_speaking
- face_visible
- head_x
- head_y
- mar_value

From the audio sensor:

- self.speech_detection.is_recording()
- self.speech_detection.get_sound_pressure_level()
- self.speech_detection.get_immediate_final_transcription()

Key methods.

Next thing to do it to look at some of the key methods in the `main.py` scrip

```
def reactWithRobotFace(self):
```

This method is used to set the current state of the robot. For starters it is either in idle mode (happy and blinking) or it is in speaking mode.

```
self.robot_face.set_state(RobotState.IDLE)
```

The set_state method in the robot face can be used to change the current robot state. It uses an enum RobotState that can be either IDLE, SPEAKING, or LISTENING.

```
self.robot_face.set_eye_position(x, y)
```

This method is used to set the position of the eyes of the robot face.

Callback methods

Robot Voice handler callback methods:

```
def handleRobotVoiceEnded(self):

def handleRobotVoiceStarted(self, text):

def handleRobotVoiceInterrupted(self):
```

These are called automatically when the **robot** either **starts speaking**, is **interrupted** or when the **speech ends** allowing you to react to these events if needed.

Speech_handler callback methods:

```
async def handle_human_speech_pause(self, transcription: str):

async def handle_human_speech_end(self, transcription: str):
```

These are called automatically when the **human** either **pauses speaking**, or when the **speech ends** allowing you to react to these events if needed.

LLM_handler callback methods:

```
def handle_LLM_chunk(self, chunk: str):

def handle_LLM_complete(self, response: LLMResponse):
```

These are called automatically when the **LLM** outputs either a small part of the answer (a chunk) or when the **LLM** is finished returning the full answer, allowing you to react to these events if needed.

Understanding all of the above methods and sensor values is a good first step to take before implementing your improvements to the robot face.

Assignment Details

Part One: Making the robot Initiate the conversation

Goal: The robot should start the interaction in a socially appropriate way.

- Use facial detection to decide when to initiate.
- Make the robot's eyes look toward the detected person to establish contact.
- Add a simple greeting once the face has been detected.

Part Two: Improving turn-taking

Currently, the robot waits for about three seconds of silence before it decides that the human has stopped speaking. This creates long pauses.

- Improve this by using the Mouth Aspect Ratio (MAR).
- When MAR shows that the user's mouth is closed and not moving, the robot can respond without waiting the full three seconds.
- Use one of the callback methods to initiate this

Part Three: Handling streaming output

Right now, the robot waits for the full transcription before generating a response. However, the method handle_LLM_chunk already receives streaming words.

- Collect the chunks in a buffer.
- As soon as you have a full sentence, send it to the speech system.
- Look for certain characters in the chunks to decide when you have a full sentence.

This will allow the robot to speak more naturally and reduce delays.

Part Four: Testing your improvements

Create an experiment that compares both the original system and your improved system with human participants.

- Each participant should try both versions (within-subjects design).
- Counterbalance the order (some participants see the original first, others see the improved version first).
- Ask participants background questions, such as their familiarity with AI or robots.
- Use the Godspeed Questionnaire to measure their impressions.
- Optionally, also use the NARS Questionnaire before testing.

Pay attention to how you introduce the robot, as context can influence results. Also make sure to note any other outside factors of- or occurrences during the experiment that could impact the participant's opinion on the interaction.

Optional Challenges

(Note: These are not required, but you might find them interesting)

For deeper exploration, you may also:

- Create a facial expression that shows the robot is listening. This animation already exists but the sprites (images) for it are just showing a blank face. If you want to create an animation, you have to:
    - Uncomment the place in the reactWithRobotFace, where the RobotState.LISTENING is set.
    - Edit the 12 frames of the animation for the eyes: (sprite_eyes_49.png to sprite_eyes_60.png) and also for the mouth: (sprite_mouth_49.png to sprite_mouth_60.png). These can be found in the /sprites_eyes and /sprites_mouth folders.
- (Advanced) Make the robot interruptible, allowing the user to cut in while it is speaking. Note that this is challenging without the user wearing headphones as the robot face easily starts thinking its own utterings are questions toward itself trigger a cycle where the robot starts talking with itself.

Appendix

Appendix 1 (below): Godspeed questionnaire
Appendix 2: (further below): The Negative Attitudes Toward Robots Questionnaire.

English

Translated by: Christoph Bartneck
Publication: https://doi.org/10.1007/s12369-008-0001-3
Instructions: Please rate your impression of the robot on these scales:

Anthropomorphism

Fake	1	2	3	4	5	Natural
Machinelike	1	2	3	4	5	Humanlike
Unconscious	1	2	3	4	5	Conscious
Artificial	1	2	3	4	5	Lifelike
Moving rigidly	1	2	3	4	5	Moving elegantly

Animacy

Dead	1	2	3	4	5	Alive
Stagnant	1	2	3	4	5	Lively
Mechanical	1	2	3	4	5	Organic
Artificial	1	2	3	4	5	Lifelike
Inert	1	2	3	4	5	Interactive
Apathetic	1	2	3	4	5	Responsive

Likeability

Dislike	1	2	3	4	5	Like
Unfriendly	1	2	3	4	5	Friendly
Unkind	1	2	3	4	5	Kind
Unpleasant	1	2	3	4	5	Pleasant
Awful	1	2	3	4	5	Nice

Perceived Intelligence

Incompetent	1	2	3	4	5	Competent
Ignorant	1	2	3	4	5	Knowledgeable
Irresponsible	1	2	3	4	5	Responsible
Unintelligent	1	2	3	4	5	Intelligent
Foolish	1	2	3	4	5	Sensible

Perceived Safety

Anxious	1	2	3	4	5	Relaxed
Calm	1	2	3	4	5	Agitated
Still	1	2	3	4	5	Surprised

The Negative Attitudes Toward Robots Questionnaire.

Item No.	Questionnaire Items	Sub Scale
1	I would feel uneasy if robots really had emotions.	S2
2	Something bad might happen if robots developed into living beings.	S2
3	I would feel relaxed talking with robots. a)	S3
4	I would feel uneasy if I was given a job where I had to use robots.	S1
5	If robots had emotions, I would be able to make friends with them. a)	S3
6	I feel comforted being with robots that have emotions. a)	S3
7	The word "robot" means nothing to me.	S1
8	I would feel nervous operating a robot in front of other people.	S1
9	I would hate the idea that robots or artificial intelligences were making judgments about things.	S1
10	I would feel very nervous just standing in front of a robot.	S1
11	I feel that if I depend on robots too much, something bad might happen.	S2
12	I would feel paranoid talking with a robot.	S1
13	I am concerned that robots would be a bad influence on children.	S2
14	I feel that in the future society will be dominated by robots.	S2

S1: Negative Attitudes toward Situations of Interaction with Robots
S2: Negative Attitudes toward the Social Influence of Robots
S3: Negative Attitudes toward Emotions in Interaction with Robots

Nomura, T., Suzuki, T., Kanda, T., & Kato, K. (2006). *Measurement of negative attitudes toward robots. Interaction Studies. Social Behaviour and Communication in Biological and Artificial Systems, 7(3), 437–454.*