HCIR: Homework 02

Names: Daniele Affinita, Klara Golubovic, K.M. Moshiur Rahman Songlap, Akhilan Ashokan

Task 1: Verbal and Nonverbal Behaviors

Task 1.1

QT robot

As it says on the website, QTrobot has over 30 facial expressions which are visible on the display at the front of his head. He can blink his eyes and raise his eyebrows, even if he is not interacting actively. Expressions such as happiness, attentiveness, and pride or embarrassment can be expressed through the screen with its displayed face. In addition to that, it can move its arms as a reaction to people's behavior or to underline its verbal speech. As his kinematic chain consists only of his head, and left and right arm, he cannot move around and stays in his place. Furthermore, he can yaw and pitch his head to interact during a conversation.
Concerning the consumption and production of its vocalics, QTrobot has a ReSpeaker V2 which includes for example a digital mic with voice activity detection and far-field voice capture to answer to verbal signals of its environment. In addition to that, it includes an Nvidia Riva and Acapella which supports text-to-speech for multilingual conversations as well as Bark/Suno which is a „fully generative text-to-audio model based on GPT style". This enables QTrobot to have also verbal conversations. As a user, you can further deploy speech-based interactions and add character to the way the robot talks to people. So, the robot can have a fluent conversation in 30 different languages and 200 voices to interact with the human as a capable conversation partner.
Used reference:

- `https://luxai.com/humanoid-social-robot-for-research-and-teaching/`

NAO robot

The NAO robot has 25 joints as well as 25 degrees of freedom to move its body parts in different directions. Looking at the head joints, the robot is among other things able to nod, pitch, or yaw its head. Having a torso, head, and right and left arm as well as a right and a left leg, the robot's body looks similar to a human body. Its predefined postures are for example sitting, crouching, or lying, which allows the robot to be flexible regarding its capabilities of body movements. Concerning facial expression, the robot has a tiny hole as a mouth which cannot move. But the robot has two LED eyes that can blink in different colors and durations. Even though the eyes are just holes in the robot's head, it can blink with those LED eyes during a conversation to express itself. If the eye color of the LEDs is blue, the NAO robot shows that he listens.
Looking at the verbal behaviors, the robot can speak in 20 languages. It has two speakers at the left and right side of its head with a broadcast system, four omnidirectional microphones, and two video cameras at the front of its head which support verbal communication.
Used reference:

- `https://www.robotlab.com/support/nao-anatomy-sensing-and-movement-on-your-robot#:~:text=The%20NAO%20robot%20has%2025,can%20occur%20on%20its%20body`
- `https://www.aldebaran.com/en/support/nao-6/3-interactions`

aibo robot

As aibo represents a dog as an animal, it does not communicate in words. aibo has seven moveable body parts with in total of 22 degrees of freedom. This allows him to move like a dog and behave according to certain situations. For example, it can crouch or sit to react to human requests. Just like a real dog, the robot can wag with its tail to express happiness and or lean its head towards the human for cuddles. The robot's eyes are OLEDs which are displays and allow aibo to blink or direct his gaze. Furthermore, to express his needs and wants, the robot closes and opens its mouth or moves its paws and ears. In general, this robot's nonverbal communication is a lot of imitation of dogs' behavior.
To communicate verbally, the robot has a speaker and four microphones to make dog sounds. aibo has voice recognition abilities to answer verbal input and can locate the source of sounds to turn its head towards it. It can make sounds in different pitches and lengths to express its behavior.

- `https://us.aibo.com/feature/feature2.html`
- `https://helpguide.sony.net/aibo/ers1000/v1/en-us/contents/TP0001970140.html`

Task 1.2

```
<bml xmlns="http://www.bml-initiative.org/bml/bml-1.0"
       character="Alice"
       id="bml1">
   <gaze id = gaze1 start = "0" target = "human" />
   <speech id="speech1" start="1">
       <text>
       Hello!
       </text>
   </speech>
   <gesture id="wave" lexeme="hello-waving" start="1" end="5">
   <speech id="speech2" start="wave:end">
       <text>
       Glad to see you!
       </text>
   </speech>
   <head id="nod" lexeme="NOD" start="gaze1:end" end="9"/>
   <posture id="happy_swirl" start="nod:end" end="12">
       <stance type="SWIRL"/>
   </posture>
</bml>
```

Task 3: Incremental Dialogue Processing in a Micro-Domain

1. What do you understand by „incremental dialog processing“?

Incremental dialog processing is used in human-human communication to describe how humans understand and produce language. In particular, it refers to the ability of humans to use their knowledge coming from different sources to decide when it is the right time to talk. In addition to that, people exchange feedback during the conversation, which helps to coordinate whose turn it is to speak. This approach is much more flexible compared with rigid turn-taking and allows people to begin speaking even though they are not sure what to say, producing utterances incrementally. Implementing this on a robot means that the processing should start before the input is finished, and an output should be available as soon as possible. Although this approach allows us to obtain a natural conversation, it is not trivially implementable on robots. Since it is not based on a predefined set of rules, it requires to

use "human common sense" which has to be formalized in robots.

2. How does incremental dialog processing support human-centeredness in interaction?

Incremental dialogue processing supports human-centeredness in interaction by aligning to a natural conversation. In particular, there are 3 main aspects that put the human at the center of the conversation process: firstly the listener doesn't always wait for a complete utterance before responding, processing information incrementally, this provides a smooth conversation; secondly, the speaker can react to mid-utterance reactions and feedback, this allows the human to control the conversation by providing some signals; thirdly when a speaker is interrupted he is able to resume the conversation from the point where he was interrupted, resulting in a meaningful conversation. So, Human-centeredness is achieved by interpreting feedback during the conversation, allowing the speaker to adapt to the listener's needs and preferences.

3. In the example presented in the paper, how did prosodic analysis contribute to incrementality?

In the given example, they used a data processor in the Sphinx frontend. The combination of incremental processing and prosodic analysis made it possible for the system to give feedback within 200 ms. For that, they used F0-extraction „ by first finding pitch candidates [...] for each audio frame using the SMDSF algorithm“ (Gabriel Skantze and David Schlangen, 2009). By parameterizing the F0 values and using a machine learning experiment on the installment-ending digit in the collected data, they found out that there is an equal amount of both types of 50.9 percent of baseline. By using incrementality the system can detect a rising or falling pitch and give an immediate mid- or end-of-sequence reaction utterance and wait if more words are following. As the paper mentions in 4.2 the system is therefore based on turn-taking decisions with a combination of ASR, prosody, and silence thresholds.

4. In the architecture for spoken dialog systems presented in the paper, why is TTS connected to Discourse Modeller?

The purpose of the Discourse Modeller is to interpret utterances considering the context to generate coherent prepositions. As additional tasks, it identifies discourse entities, resolves anaphora, and keeps track of the grounding status of concepts. To do so the Discourse Modeller engine has to maintain an internal state. The Discourse Modeller engine used in the paper is *GALATEA* which models utterances coming from both the user and the system itself. Since the system utterances are produced by the Text To Speech engine, the diagram connects the TTS module with the Discourse Modeller (DM) so that the output produced can be considered by the DM.

Feedback

1. How much time did you spend on doing this sheet per person? Anonymize your answer!
In total, we spent 8 hours on this homework, and on average each person took about 2 hours. To add to that, it was difficult to divide the homework among four people so we can only give an approximate number of hours here.
2. Was this sheet too easy / easy / ok / hard / too hard?
Some of the questions of task 3 (e.g. 3. question) were difficult as the paper is quite long and sometimes complicated to understand. There were many new terms and concepts that we needed to understand

before actually working on the questions. For the coding task, it depends on the knowledge of each person to evaluate whether the tasks were hard or not. Two of us have a little bit of experience in coding and thought that this task was very hard.
3. What additional resources (blogs, papers, books, tutorials, etc.) did you use? Please provide links or references.
For task 1 we used some new references which are given below each paragraph to elaborate further on the verbal communication. For example, for the robot aibo it was hard to find enough details about certain capabilities and the implementation of communication. For the coding task, the qibullet documentation was used.
4. Did you face any issues while solving this sheet?
It was hard to divide the work among several people and as mentioned in the second point, the paper was sometimes complicated to understand.
Furthermore, the homework is very extensive, even for four people. Because of that, it is not always possible for everyone to understand each other's work. Especially for the coding tasks, often there is no time for everyone to understand the whole code as only one or two students work on it.