Intention and Engagement Recognition for Personalized Human-Robot Interaction, an integrated and Deep Learning approach

Suraj Prakash Pattar, Enrique Coronado, Liz Rincon Ardila, Gentiane Venture

***Abstract*— The quality of the interaction between two individuals depends upon not only exchange (i.e. understanding partner's intention and reacting to it), but also on how personalized is the interaction. In this work, we have set out to accomplish these objectives for Human Robot Interaction. For this, we have developed a distributed and multimodal data acquisition and interaction manager architecture aiming to enable personalized Human-Robot Interactions. In the proposed approach, high-level perceptual capabilities (i.e. recognizing human activity and engagement) are performed by an Autoencoder, which is a Deep Learning and Unsupervised Learning method. This Autoencoder module is integrated with a facial recognition and a dialog manager (speech recognition and speech generation) to enable personalized interaction. We discuss the advantages of Autoencoders over Supervised Learning methods, and how our proposed architecture can be used to increase the duration of interaction with a robot during unscripted scenarios. Experimental validations are also performed in real Human-Robot interactions using a humanoid robot.**

***Index Terms*— Personalized Human Robot Interaction, Intention and engagement recognition, Deep learning, Autoencoder, Intelligent and autonomous robots.**

I. Introduction

In the past decades, robots have been serving in strictly industrial and professional tasks (e.g. pick and place, product inspection, welding, search and rescue and underwater exploration) [1]. In most of the cases, these robots were isolated from humans only allowing very basic ways of interaction, such as the use of keyboard, teach pendant or remote controller interfaces. However, the recent integration of robots in "social" environments requires more usable ways of communication [2]. This can be done by the development of system architectures that integrate those sensory and perceptual systems used by humans to enable "natural" communication between them [2]. Moreover, to enable effective and valuable communication these robots must be able to initiate and maintain personalized interactions [3], [4].

In an effort towards the development of social intelligent and autonomous robotic systems able to perform personalized Human-Robot Interactions (HRI) this paper presents a multimodal data acquisition system and interaction manager architecture, aiming at enabling the training of machine learning algorithms, as well as the on-line validation of these algorithms. This data acquisition system is based on state-of-art methods for skeleton recognition, face recognition and natural language processing. We also present a classifier capable of detecting the intention to engage. This classifier is based on an unsupervised deep learning approach denoted as autoencoder. To the best of the knowledge of the authors of this article, this approach has not been applied before to enable engagement recognition.

All the authors are with the Department of Mechanical Systems Engineering, Tokyo University of Agriculture and Technology, 2-24-16 Nakacho, Koganei, Tokyo, Japan. Corresponding author's email: pattarsuraj@gmail.com.

II. Related work and positioning

Engagement is an important social ability that can be used to improve the quality, usability and acceptability of social robots. The term "engagement" is recognized as a complex concept that admits several definitions in the Human-Computer Interaction (HCI) literature [5], [6]. In the domain of social robots it is possible to define engagement as "the process by which two (or more) participants establish, maintain and end their perceived connection during interactions they jointly undertake" [3], [4].

The task of detecting engagement is often defined as a machine learning problem. Classical input features for those machine learning methods aiming at detecting engagement have been mainly based on spatial information. Examples of this type of data are human position, velocity, head pose, gaze and facial expressions [7]. Most works report the use of this data on supervised classifiers. Examples are Hidden Markov models and Conditional Random Fields which were used in [8], and Dynamic Bayesian Networks and Support Vector Machines (SVM) which were used in [9]. A more recent trend in machine learning which has dramatically improved the state-of-the-art on areas such as speech recognition and visual object recognition is the use of Deep learning approaches [10]. However, these approaches have been poorly explored for engagement recognition. Classical and most successful Deep learning models are Recurrent Neural Network (RNN) [11] and Convolutional Neural Networks (CNN) [12], which are also supervised methods.

Our approach differs from most of the works reported in literature for intention and engagement classifications, which make use of mainly classical machine learning and supervised methods. Instead, we make use of a deep Autoencoder, which is a deep learning method generally used to learn efficient data coding in an unsupervised manner [13]. Advantages of Autoencoders are as follows: i) Regular Deep Learning Networks using CNNs and RNNs need a balanced labeled dataset for effective classification, which is not the case with Unsupervised Learning. The dataset is bounded to become unbalanced as we succeed in increasing the duration of interaction; ii) Once we have validated a model making use of labeled dataset, there is no further need to label the input data which is a time-consuming process; iii) For future cases, when there is a new behaviour to be identified, it is

much easier for an unsupervised model to detect it as an anomaly than for a supervised model to classify it. It also deals well with cases where one has an imbalanced data-set.

III. Multimodal Data Acquisition and Intention/Engagement Recognition System

The proposal for a multi-modal data acquisition and interaction system is represented in Fig. 1. The system is divided into three main sections as follows:

- **Human Input**: represents the human motion and activities before/during/after the interaction with the robot.
- **Intention/engagement classifier**: represents the skeleton data acquisition part of the system. This data is labeled and used to create an autoencoder for further classification of the human interaction behavior.
- **Integrated Interaction**: represents our proposal to make the robot recognize and react with personalized interactions. It includes the devices and processing used for a personalized interaction with the human. This system includes a web-camera for facial recognition, a microphone for speech recognition, a dialog manager, a tablet for displaying dialogues and a humanoid robot that interacts with the humans.

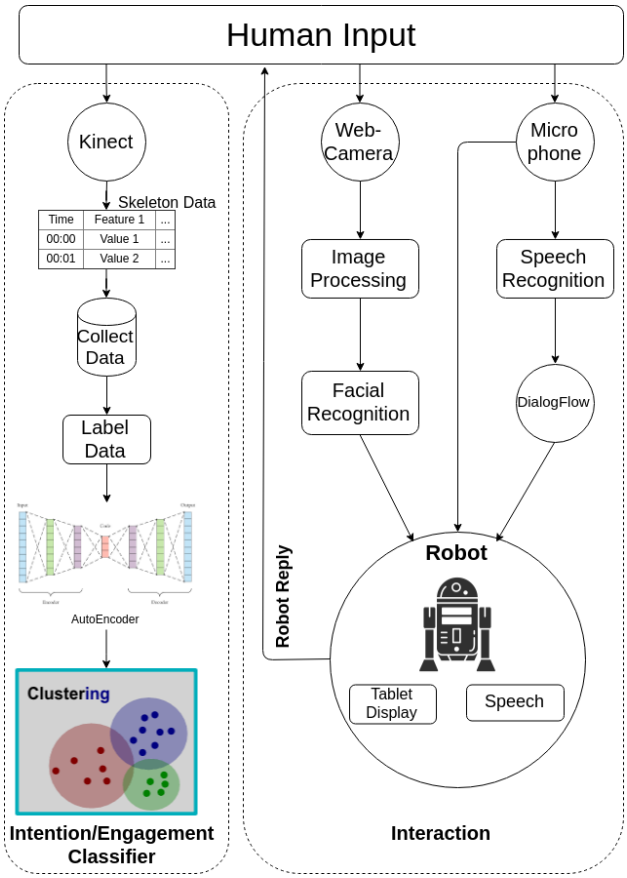


Fig. 1: Our proposal for intention and engagement recognition for HRI: Overview of a general setup

A. The facial interaction (facial recognition system)

By using facial recognition the robot is able to adapt its behaviors based on the recognized users. The procedure applied to perform face recognition consists of three steps: *face detection*, *face verification* and *face classification*.

In the *face detection* step, the algorithm explores an image and decides if there is any face in that picture. It is done by segmenting and separating the faces from other background objects. For this step, we use a method based on Histogram of Oriented Gradients (HOG) [14], which is implemented in off the shelf in [15].

In the *face verification* step which is essentially a 1:1 problem, where one recognized face must be associated with a specific Name/ID. This process requires to deal with cases where the faces are at different angles or partially turned away. For this, each image is wrapped such that the eyes and lips are always in the same place in the image. To achieve this, a method to estimate face landmarks (i.e. specific points that exist on each face) is needed. However, this process could be highly costly in terms of processing time as one can have a database of N number of people. This step is a 1:N problem. In order to reduce computational cost, the algorithm needs to use only few basic landmarks from each face. To find which measurements are most important Deep Learning techniques are applied.

A Deep Convolutional Neural Network is trained to generate a specific number of landmarks for each image. The training process for this Deep Learning model is described below: i) Provide an image with a face of a known person (Image-1), ii) Provide another image of the same person (Image-2), iii) Provide an image of a completely different person (Image-3). The algorithm tweaks the weights of the Neural Network such that the measurements generated for Image-1 and Image-2 are closer compared to the measurements generated to that of Image-2 and Image-3. As these steps are repeated for different quantity of images. The network generates the specified number of measurements for each person. These measurements are designated as *embeddings* in this machine learning. Now, the pre-trained model is used, and the own models are trained on top of this with our own images data set. This is described as the *Transfer Learning*.

Finally, the *face classification* step is required to find the image, in our registered user database with the closest measurements to our test image. This can be achieved with any basic machine learning classification algorithm. In our work, we have used of One-Shot Detection algorithm and K Nearest Neighbours (KNN).

B. The dialog manager (speech recognition)

The speech plays an important role in the interaction between humans. The Pepper robot presents a natural affordability of speech interaction i.e. due to its humanoid features when users see the robot they naturally assume that they can speak with it. Due to this, we built a speech interaction subsystem shown in Figure 2. The processes involved in this sub-system are described bellow:

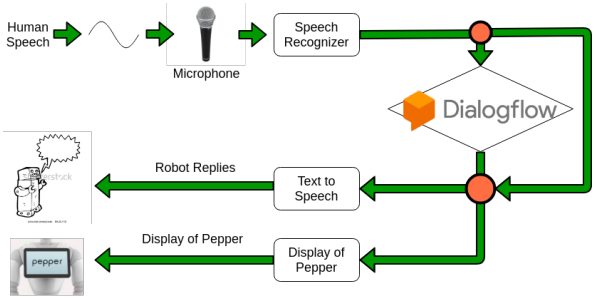
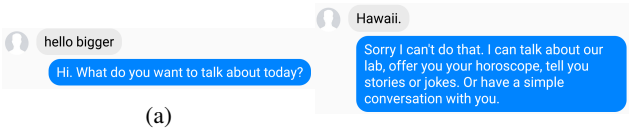


Fig. 2: Speech Recognition to Dialogflow Interaction

Dialog interaction: An external microphone device is used to perform the data acquisition in the Pepper robot, due to the best quality of the sound with this device compared to the microphone already installed in the robot. This sound data is processed and applied to recognize the speech by the Speech Recognizer module. For that, the python speech recognizer library [16] is used.

Interaction with a dialog displayed by the tablet interface: In order to provide feedback to the user, the output of the Speech Recognizer and the robot's speech are both displayed on a tablet attached to the robot. An example of the provided interface is shown in Figure 3 where the text marked in gray bubble is the "Live-Subtitle" of what the robot "understood", and the text in blue bubble shows the response of the robot. This figure presents how the robot deals with errors in the speech recognition. On the one hand in figure 3 (a) the user intended to say *"Hello Pepper"*, but the Speech Recognition using Google Speech Recognition as backend interpreted it as *"Hello Bigger"*. However, the robot was able to "understand" the general meaning of the phrase and responds correctly. On the other hand in figure 3 (b) the speech recognizer's output was not understood by the dialog manager at all. In this case, the robot responds with a clear message that it does not understand and explains what it is capable of talking about.



(a) (b)

Fig. 3: Examples of the dialogue feedback to the users with two different types of failure (a) that doesn't affect the flow; (b) that affects the flow of interaction

Chatbot Integration: In order to understand the general meaning of the sentences of the users we use a Natural Processing Language tool that enables the development of advanced chatbot interfaces. This tool denoted as Dialogflow acts as the main engine for our Speech Interaction section.

Figure 4 represents the Dialogflow sections as: **Invocation**, **User Says**, **Responses** and **Entities**. **Entities** are useful in cases where we need to gather information from external sources using a WebHook. These services can be related to weather, time, finance, etc. The Dialogflow agent is

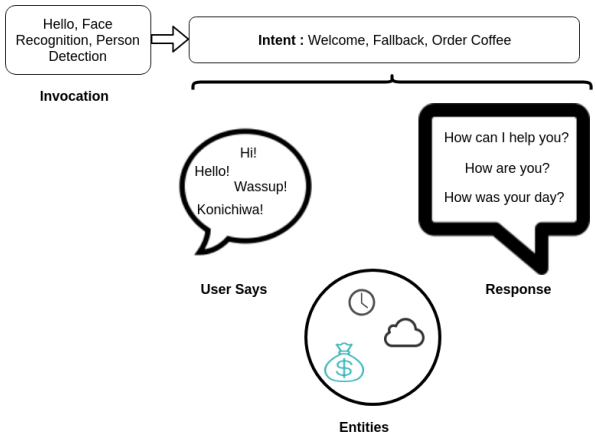


Fig. 4: Structure of the Dialogflow functionalities

activated with an **Invocation**. This invocation can be the Face Recognition, Person Detection or using a *Hot Word* for activating the system. In our scenario, it includes Face Recognition or a simple greeting from the user. Once the agent is invoked, the user input is matched with any of the numerous **Intents**. For example, for a general greeting, the intent matches the user speech with its **User Says** section which contains various greetings or if it is closely related to the examples in this section. If matched, it provides randomly one of the responses recorded in the **Responses** section. Another example is shown in Figure 3 (a), where the recognized speech is matched with an appropriate **Intent** defined as: *smalltalk.greeting.hello*. In cases where the user speech is not recognized, the Dialog Manager sends it to a default **Fallback Intent** where the agent replies with a default response that it does not understand what the user is trying to say or it does not have the capability to complete the user's request and specifies what it can do. An example of this case is shown in figure 3 (b).

C. Features for intention and engagement recognition (Skeleton tracking)

For *Engagement Classification*, we refer to the ranking of features provided in [17]. We also use the same proposed device, Kinect version 2.0, which is a video game oriented camera widely used in robotics research to perform 3D scanning of environments and 3D skeleton tracking. However, this last feature is only available in the official Software Development Kit (SDK), which is only Windows supported. Taking inspiration from [17], we made use of the features shown in Table I in our work.

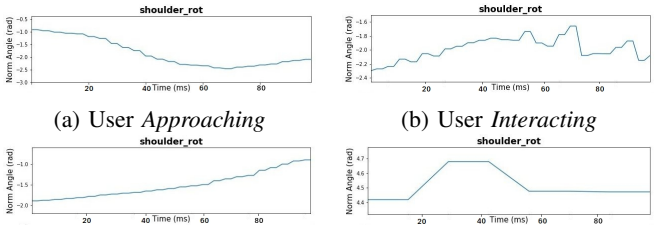
In total we have 55 features. We use these features for the training of the proposed Engagement Classifier Model. Data collected, it is hand-labeled. For this process, we match the time-stamp from the Kinect sensor data and the time-stamp on the video camera. Due to this unbalanced dataset, it can be difficult for the models to recognize patterns of the underrepresented classes. Although there are techniques available for dealing with such imbalanced dataset such as SMOTE (Synthetic Minority Over-sampling Technique),

No.	Name	Unit	Description
1	face_engaged	[0; 1]	If user is looking at Kinect
2	face_glasses	[0; 1]	If user is wearing glasses
3	face_happy	[0; 1]	If user is smiling
4	face_lefteyeclosed	[0; 1]	If left eye is closed
5	face_righteyeclosed	[0; 1]	If right eye is closed
6	face_lookingaway	[0; 1]	If user is looking away from Kinect
7	face_mouthmoved	[0; 1]	If user's mouth is moving
8	face_mouthopen	[0; 1]	If user's mouth is open
9	face	rad	pitch, roll, yaw angles
10	shoulder_rot	rad	Rotation of the shoulder
11	elbow_left	m	x, y, z positions
12	elbow_right	m	x, y, z positions
13	head	m	x, y, z positions
14	hip_left	m	x, y, z positions
15	hip_right	m	x, y, z positions
16	neck	m	x, y, z positions
17	shoulder_left	m	x, y, z positions
18	shoulder_right	m	x, y, z positions
19	spine_base	m	x, y, z positions
20	spine_mid	m	x, y, z positions
21	spine_shoulder	m	x, y, z positions
22	wrist_left	m	x, y, z positions
23	wrist_right	m	x, y, z positions
24	pedes_pos	m	x, y, z positions
25	time_stamp	ms	Kinect Device Time

TABLE I: List of features used and their descriptions

under-sampling majority classes and over-sampling minority classes, we risk losing useful data for effective classification of data.

To deal with such imbalanced dataset, we proceed with Unsupervised Deep Learning. In particular, we make use of Autoencoder to learn efficient embeddings in our data. Once data-labeling is completed, we proceeded with **Data Visualization**, to observe if we could spot some variations in the data for our four classes, namely: *Approaching*, *Interacting*, *Leaving* and *Uninterested*. An example is shown in Figure 5. Before visualization, the features are first normalized.



(a) User *Approaching* (b) User *Interacting*

(c) User *Leaving* (d) User *Uninterested*

Fig. 5: Data visualization of the human shoulder rotation from Kinect

IV. The intention and recognition classifier, training and classification

For the intention and engagement recognition an Autoencoder model was created. Due to the unsupervised nature of this method, the input data does not require any labels while training. Instead in the output layer, the autoencoder tries to generate data that closely matches the input. In this work, the model is trained on high dimensional human skeleton data. Then we use the latent compressed data in latent space for visualization. In future work, the decoder part would be chopped off and the encoder would preserved for further use as input to a clustering algorithm.

Autoencoders generally consist of two parts: an encoder and a decoder. The encoder reads the input data and compresses it to a compact representation. The decoder reads this representation and recreates the input. The whole process is to learn the identity function with minimum reconstruction error. Once the model is trained with minimal reconstruction error, the latent space i.e. the compact representation of the input is used for the classification.

Network Architecture: Our network is constituted by the following: a) The encoder is a RNN that takes a sequence of input vectors; b) The encoder to latent vector is a linear layer that maps the final hidden vector of the RNN to a latent vector; c) The latent vector to decoder is a linear layer that maps the latent vector to the input vector for the decoder; d) The decoder is a RNN that takes this single input vector and maps to a sequence of output vectors.

V. Experimental setup

The robot used for the implementation of the system is a Pepper robot and it was equipped with the following extra devices to extend the capabilities for the complete engagement interaction as shown in Figure 6.

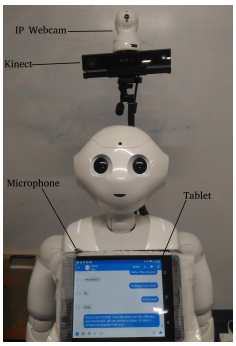


Fig. 6: Experimental Setup with Pepper Robot

The main data collection device i.e. the Kinect Sensor is placed right behind the Pepper robot's head; the IP Webcam is placed on top of the Kinect sensor which is used for Facial Recognition. The android tablet is fixed on top of Pepper's inbuilt tablet. The microphone is also attached to the same tablet, it is connected to the Laptop with *i3* processor running Linux OS. The IP Webcam is also connected to the same Laptop through either Wi-Fi or Ethernet cable. The Pepper robot's *Text to Speech* or *Animated Text to Speech* receives its commands from the same Laptop device. The Kinect device is connected to another Laptop with *i7* processor running Windows 10 OS. This was necessary as the Kinect v2 requires Windows 10 OS with a dedicated USB-3 port. The communication between all these devices is accomplished using NEP [18], which is a cross-platform robot programming framework we developed to enable inter-process communication between nodes in some of the most popular programming languages and operating systems by

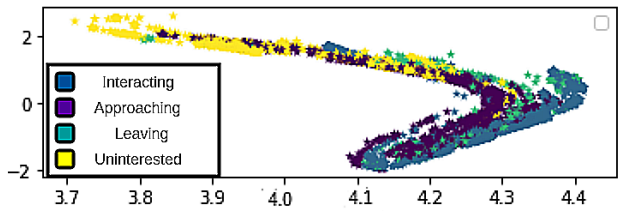


Fig. 7: Autoencoder for the recognition during the HRI integrated in the proposed overall system (with PCA: 55-D input features compressed to 2-D for visualization)

using different middlewares such as ROS and ZeroMQ. There is another digital camera to record the interactions for further analysis.

The subjects of the experiment had the following characteristics: English speaking; Aged between 20 to 30; Able bodied; Persons studying or working on Robotics.

VI. Results and Discussion

A. Facial recognition

In the initial experiments we used One-Shot Detection as it would be faster to add new persons to our database, however its accuracy was limited to apply for the face recognition.

We implemented the K Nearest Neighbours (KNN) algorithm. To proceed with KNN, we collected image-data of all the lab-members. The images were taken from a cell-phone camera in burst-shot mode such that 100 images each were taken quickly for each subject. In data collection one needs to be careful and foresee the environment where the actual prediction would take place. Since the early tests were being held in the same place, we collected the images in the place itself with all the reflective surfaces present. Also, the frame of taking images is important since the Pepper robot is much shorter than average humans. For the Face Recognition model, we used the external IP Webcamera. The use of this camera provides flexibility to position it as per our requirements. It means that there are part of the interactions where the robot moves its head away from the user, or to a general direction that is meant to be facing. This can cause problems for future interactions if the robot head is not re-positioned each time. In addition, the use of an IP webcamera using the tools such as OpenCV provides us the flexibility to be robot independent. Also this hardware is more accessible in terms of data streaming than the hardware on the robot. For the interaction classifier, the autoencoder learns the identity function, so the sequence of input and output vectors must be similar. Every output is a tuple of a mean μ and standard deviation σ. Let this μ and σ parameterize a Gaussian distribution. Now, we minimize the log-likelihood of the input under this distribution. We trained this model using backpropagation into the weights of the encoder, decoder and linear layers [19]. We run the recurrent Autoencoder with a 20-D latent space i.e. the 55-D input features are compressed to 20-D by the Autoencoder. These 20 dimensions are plotted to 2-D after applying PCA to compress them further for visualization purpose only. In the Figure 7, the different classes can be easily visualized with PCA. We can also notice that some of the data points from *Leaving* and *Approaching* classes are presented in the *Interacting* cluster. This can be due to the resolution of the time-stamp when hand-labeling the data. As the time-stamp of video recording only had resolution up-to seconds, some of the data for *Interacting* class might have been wrongly classified as *Approaching* or *Leaving*.

B. Interaction Time Comparison

After executing the experiments of the detection of engagement, we now explore how we approached increasing the duration of interaction. For the experiment interactions, the participants were not given explicit directions on how to interact with the robot as we wanted to record natural interactions. One small direction given to the participants was to approach the robot and say a greeting for example, "Hello" in cases where the Face-Recognition block was not active. The training set comes from the data collected by the Kinect sensor and video recording of experiments with a camera. The time stamps on the camera and Kinect sensor were synchronized. We obtained time-series skeleton data from the Kinect sensor and it was hand-labeled using the video recordings. We recorded 4 interactions *without* the tablet display, and 6 interactions *with* the tablet display to see its impact on maintaining the interaction. As we only had a limited number of English speaking participants, the experiments could not be repeated without giving away the novelty of first time interaction. We compare only the mean interaction time between the two groups, and change in the interaction time for the two participants *B and D* who participated for both scenarios. Figure 8 and 9 represent the interaction time with and without tablet display. Within the two graphs, with the tablet display, the interactions lasted longer. As we can see, from participant ID **B** and **D** who participated in both scenarios, the interaction times nearly doubled when Tablet modality was introduced. It was reportedly due to the help provided by the display to clearly indicate the robot's current status and *"live subtitles"* shown in real time. This clearly indicates that using the tablet as an additional mode of interaction helps for increasing the interaction time when the speech recognition and dialogue generation have a significantly high number of failures.

Also in Figure 9 we notice participant IDs **G** and **H** interacted considerably longer than other participants. This could be due to the fact that the two participants had more frequent physical interactions with robots in their daily lives. As the chatbot, explained in sub-section III-B, was improved iteratively after each interaction and these two interactions were chronologically the last two interactions, it could signify that the chatbot's overall conversational abilities were better in latter stages.

C. Observations and discussion

Some important points for the intention and engagement recognition described by the users as shown below:

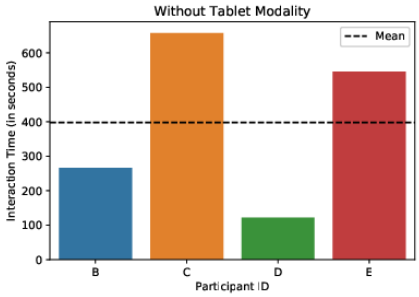


Fig. 8: Interaction time without tablet modality

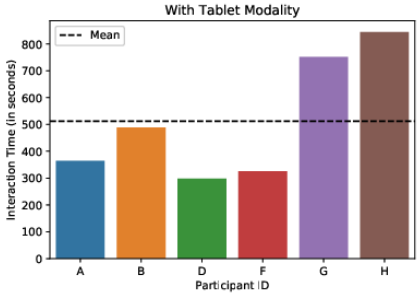


Fig. 9: Interaction time with tablet modality

Interaction time: There was a considerable importance in the time delay during the interactions as observed and reported by the users. These delays were mainly caused by the robot hardware and internet connection which was essential for the speech interaction.

Robot feedback response for the user (tablet display): As reported by the users, they had to depend a lot on the feedback from the display to effectively interact with the robot. For example, during a delayed response due to low internet connection or faulty speech recognition, the tablet display would notify the user to wait as processing is undergoing. In the absence of this modality, the user would often get frustrated and be unaware if the robot had recognized their speech or not. The tablet display was mainly used as there is a lack of universal indicators to show the present status of the robot. As seen in our experiments, the robot visual feedback helped to increase the duration of the interaction.

Robot personalized with face recognition: The facial recognition is a crucial part in creating a personalized interaction with the user. During the experiments, it was clearly observed that upon successful facial recognition by the robot, the users reacted in a positive manner often praising it for recognizing them correctly, which impacted on the overall interaction.

VII. Conclusions and Future Work

The multimodal data acquisition and intention/engagement recognition system for HRI was developed and successfully implemented with the Pepper robot. The system was tested with users and their engagement was measured during a free interaction with the robot using verbal communication. The proposed architecture is constituted by human inputs, intention/engagement classifier and integrated interaction. In order to classify and analyse the intention/engagement an Autoencoder was modeled and implemented. The most important points from the users evaluation were noted as being the interaction time, the robot feedback to the user, and the robot personalization with the face recognition.

In future work, the system will be extended in the classifier model as a feedback response involving a decision making module for a more personalized interaction.

References

[1] S. Thrun, "Toward a framework for human-robot interaction," *Human-Computer Interaction*, vol. 19, no. 1, pp. 9–24, 2004.

[2] E. Coronado, J. Villalobos, B. Bruno, and F. Mastrogiovanni, "Gesture-based robot control: Design challenges and evaluation with humans," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2761–2767.

[3] L. Devillers and G. D. Duplessis, "Toward a context-based approach to assess engagement in human-robot social interaction," in *Dialogues with Social Robots*. Springer, 2017, pp. 293–301.

[4] C. L. Sidner, C. Lee, C. D. Kidd, N. Lesh, and C. Rich, "Explorations in engagement for humans and robots," *Artificial Intelligence*, vol. 166, no. 1, pp. 140–164, 2005.

[5] L. J. Corrigan, C. Peters, G. Castellano, F. Papadopoulos, A. Jones, S. Bhargava, S. Janarthanam, H. Hastie, A. Deshmukh, and R. Aylett, "Social-task engagement: Striking a balance between the robot and the task," in *Embodied Commun. Goals Intentions Workshop ICSR*, vol. 13, 2013, pp. 1–7.

[6] N. Glas and C. Pelachaud, "Definitions of engagement in human-agent interaction," in *International Workshop on Engagment in Human Computer Interaction (ENHANCE)*, 2015, pp. 944–949.

[7] M. P. Michalowski, S. Sabanovic, and R. Simmons, "A spatial model of engagement for a social robot," in *9th IEEE International Workshop on Advanced Motion Control, 2006.* IEEE, 2006, pp. 762–767.

[8] M. E. Foster, A. Gaschler, and M. Giuliani, "Automatically classifying user engagement for dynamic multi-party human–robot interaction," *International Journal of Social Robotics*, vol. 9, no. 5, pp. 659–674, 2017.

[9] Y. Feng, Q. Jia, M. Chu, and W. Wei, "Engagement evaluation for autism intervention by robots based on dynamic bayesian network and expert elicitation," *IEEE Access*, vol. 5, pp. 19 494–19 504, 2017.

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[11] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 2013, pp. 6645–6649.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[13] C.-Y. Liou, W.-C. Cheng, J.-W. Liou, and D.-R. Liou, "Autoencoder for words," *Neurocomputing*, vol. 139, pp. 84–96, 2014.

[14] B. Li and G. Huo, "Face recognition using locality sensitive histograms of oriented gradients," *Optik-International Journal for Light and Electron Optics*, vol. 127, no. 6, pp. 3489–3494, 2016.

[15] ageitgey, "The world's simplest facial recognition api for python and the command line," https://github.com/ageitgey/face_recognition, 2018, online; accessed 30-April-2018.

[16] A. Z. (Uberi), "Speech recognition pypi," https://github.com/Uberi/speech_recognition, 2018, online; accessed 12-June-2018.

[17] D. Vaufreydaz, W. Johal, and C. Combe, "Starting engagement detection towards a companion robot using multimodal features," *Robotics and Autonomous Systems*, vol. 75, pp. 4–16, 2016.

[18] E. Coronado, F. Mastrogiovanni, and G. Venture, "Design of a human-centered robot framework for end-user programming and applications," in *ROMANSY 22–Robot Design, Dynamics and Control*. Springer, 2019, pp. 450–457.

[19] R. Romijnders, "Auto encoder for time series," 2018. [Online]. Available: https://github.com/RobRomijnders/AE_ts