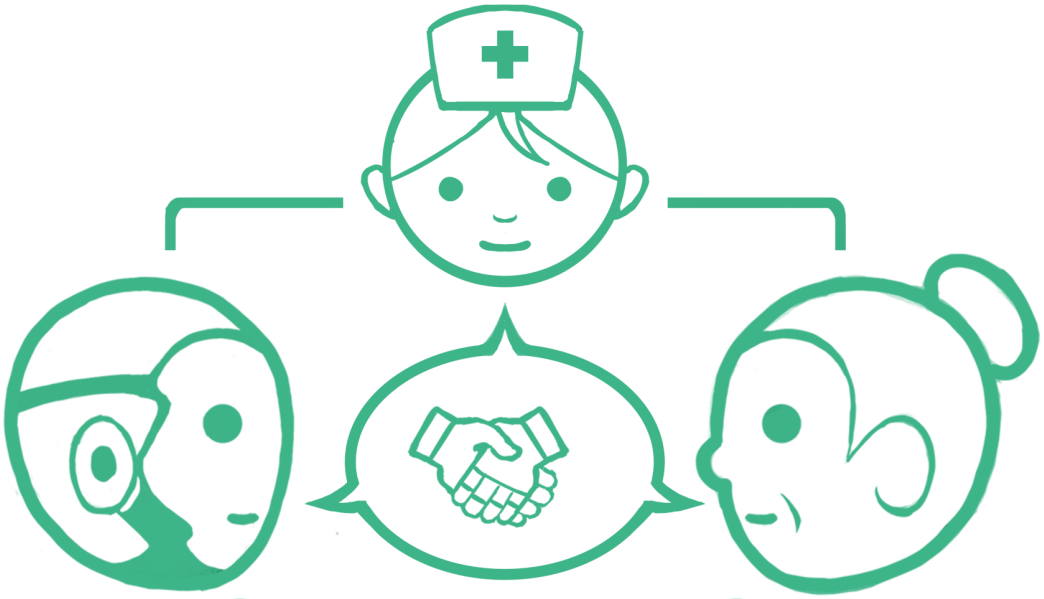


Deliverable D5.2: Multi-Party ASR and Conversational System in Realistic Environments

Due Date: 28/02/2022

Main Author: S. Gannot (BIU) and D. Hernandez Garcia (HWU)

Contributors: C. Dondrup (HWU), P. Tandeitnik (BIU), Y. Ellinson (BIU) and O. Cohen (BIU)

Dissemination: Public Deliverable

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

DOCUMENT FACTSHEET

Deliverable	D5.2: Multi-Party ASR and Conversational System in Realistic Environments
Responsible Partner	BIU
Work Package	WP5: Multi-User Spoken Conversations with Robots
Task	T5.3: Multi-party Conversational System
Version & Date	28/02/2022
Dissemination	Public Deliverable

CONTRIBUTORS AND HISTORY

Version	Editor	Date	Change Log
1	BIU	05/05/2022	First Draft
2	HWU	26/05/2022	Second Draft
3	BIU	03/06/2022	Third Draft
4	BIU	28/06/2022	Fourth Draft

APPROVALS

Authors/editors	PARTNERS
Task Leader	C. Dondrup (HWU)
WP Leader	C. Dondrup (HWU)

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

Contents

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

Executive Summary

Deliverable 5.2 reports the initial progress on task T5.3 on Multi-party Conversational System. The goal of task 5.3 is developing the multi-party conversational system for SPRING. This deliverable provides the preliminary software package for multi-party ASR with speech enhancement algorithms T3.2 & T3.3, and conversational system.

As per European Commission requirements, the repositories [11] will be available to the public for a duration of at least four years after the end of the SPRING project. People can request access to the software to the project coordinator at `spring-coord@inria.fr`. The software packages use ROS (Robotics Operating System) to communicate with each other and with the modules developed in the other workpackages.

This document will explain:

1. Developed software modules for ASR, and speech enhancement algorithms
2. Evaluation of Automatic Speech Recognition (ASR) solutions
3. Evaluation of speech enhancement algorithms: Mixture of Deep Experts (MoDE)
4. Data collection for training the multi-party conversation system

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

1 Introduction

The overall objectives of WP5 (Multi-User Spoken Conversations with Robots) are to develop a) techniques for multi-user conversation involving a robot and multiple humans, and b) the overall robot task planning.

This deliverable presents initial progress on task T5.3, Multi-party Conversational System, with the goal of developing a multi-user conversational system trained on collected data for SPRING. In this task, we will develop the multi-party conversational system, consisting of NLU, NLG, and the dialogue management modules. They will be incrementally trained on a combination of anonymous textual data collected in T1.2 and in realistic environments through the initial system of Deliverable D5.1 [7]. The conversational system will take as input a synthesis of the ongoing interaction and provide the appropriate text utterances for speech synthesis.

Deliverable 5.2 reports the initial progress of task T5.3 but does not include a full version of the dialogue system for multi-party interaction which is due for D5.4. This deliverable provides the preliminary software package for multi-party ASR with speech enhancement algorithms of T3.2 & T3.3, and the conversational system in Section 4 enabling us to start work on the multi-party dialogue system.

Section 2 describes the speech enhancement algorithms and present the evaluation of such algorithms with the ASR for improving performance.

Section 3 describes a data collection design for eliciting complex and natural multi-party conversations with a Social Robot for training the multy-party conversation system and develop the Conversational content generator (CCG) as part of the software for generating multi-party situated interactions for T6.2 (see Deliverables D6.2 [8]).

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

2 The audio processing pipeline

Here we will describe the audio pipeline from the sound captured by the ReSpeaker sound card, through the enhancement algorithm, the automatic speech recognition (ASR) module, and, finally, the dialogue system.

In this deliverable, we will focus on a deep neural network (DNN)-based single-microphone noise reduction algorithm, dubbed mixture of deep experts (MoDE) [2]. We will also report on the experiments carried out with several commercial ASR algorithms in noisy and reveberant environments. In the next steps, we will incorporate a speaker separation algorithm (using either multi- or single-microphone) to facilitate a fluent dialogue between the robot and the human, even in concurrent speakers scenarios.

2.1 Mixture of Deep Experts

Let $x(t) = s(t) + n(t)$ be the signal captured by a microphone (embedded in ARI), with $s(t)$ the clean speech signal and $n(t)$ the noise signal. Note that under this model we do not explicitly address reverberation and $s(t)$ may denote the reverberant speech as received by the microphone. As the expected reverberation time in the target room is very high, a dereverberation algorithm may be required [3, 12].

Define $\bar{\mathbf{x}}, \bar{\mathbf{s}}, \bar{\mathbf{n}}$ and $\mathbf{x}, \mathbf{s}, \mathbf{n}$ the spectral and log-spectral vectors, respectively, with components ($k = 0, 1, \ldots, L/2$):

$$\bar{x}_k = |X(k)|,\ \bar{s}_k = |S(k)|,\ \bar{n}_k = |N(k)| \tag{2.1}$$

$$x_k = \log|X(k)|,\ s_k = \log|S(k)|,\ n_k = \log|N(k)|\,. \tag{2.2}$$

Now, define the ideal ratio mask (IRM):

$$\mathrm{IRM}_k = \left(\frac{|\bar{s}_k|^2}{|\bar{s}_k|^2 + |\bar{n}_k|^2}\right)^{\gamma} \tag{2.3}$$

where γ is commonly set to $\gamma = 0.5$. Finally, define $\rho_k \in [0, 1]$, the IRM estimate, given the noisy speech utterances. For the estimation we utilize the current noisy frame and some context frames, imposing a short processing latency.

The IRM estimate, $\boldsymbol{\rho}$, is applied in the log-spectral domain to obtain an estimate of the log-spectrum of the clean speech signal.

$$\hat{\bar{\mathbf{s}}} = \bar{\mathbf{x}} \odot \exp\{-(\mathbf{1} - \boldsymbol{\rho}) \cdot \beta\}. \tag{2.4}$$

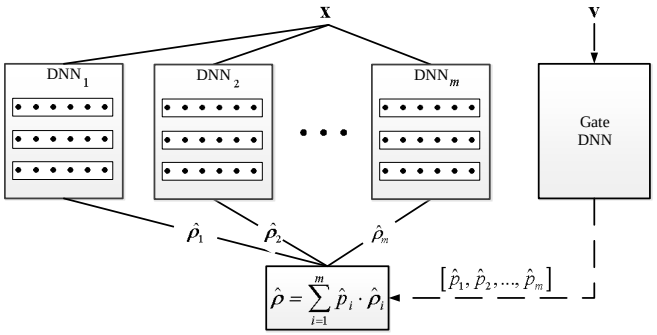
where β is the, possibly frequency-dependent, maximum attenuation level, set to trade-off between noise reduction and speech distortion (namely, the aggressiveness of the algorithm), and $\odot$ is the Hadamard product.

The gist of the MoDE algorithm is to split the enhancement task into smaller sub-tasks. Each sub-task is responsible for enhancing a specific component of the speech signal. In [1], we proposed a mixture of phonemes model, where each sub-network is responsible for enhancing a phoneme, and a 'gate' network determines the importance of each sub-network to the overall speech enhancement. In the MoDE algorithm [2], we substitute the phoneme-oriented sub-networks with 'experts' responsible for speech building blocks that are inferred in an unsupervised manner. The algorithm architecture is depicted in Fig. 2.1a. The system comprises m experts, each of which provides an IRM estimate, $\hat{\boldsymbol{\rho}}_i(n) = p(\boldsymbol{\rho}(n)|z(n) = i, \mathbf{x}(n); \boldsymbol{\theta}_i)$, $i = 1, \ldots, m$ with $\boldsymbol{\theta}_i$, $i = 1, \ldots, m$ the parameter set of the i-th expert, and $p_i(n) = p(z(n) = i|\mathbf{v}; \boldsymbol{\theta}_g)$ is the gating soft decision, with $\boldsymbol{\theta}_g$ its corresponding parameter set. Define $\Theta = \{\boldsymbol{\theta}_g, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m\}$ as the set of all network parameters to be inferred in the training stage.

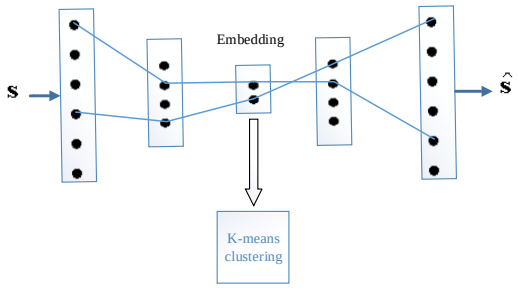
The network is trained using pairs of input feature vectors $\mathbf{x}$ and their corresponding labels, $\boldsymbol{\rho}$ and minimizing the following cost finction:

$$L(\Theta) = -\sum_{n=1}^{N} \log\left(\sum_{i=1}^{m} p_i(n) \exp(-d(\boldsymbol{\rho}(n), \hat{\boldsymbol{\rho}}_i(n)))\right) \tag{2.5}$$

with $d(\boldsymbol{\rho}(n), \hat{\boldsymbol{\rho}}_i(n)) = \frac{1}{2}\|\boldsymbol{\rho}(n) - \hat{\boldsymbol{\rho}}_i(n)\|^2$. We used a pre-training stage to initialize the system. In [1] we could use the phoneme labels for pre-training. For the MoDE algorithm, the labels are acquired in an unsupervised manner by applying a clustering algorithm to the clean speech utterances. The clustering is used to find m different patterns of

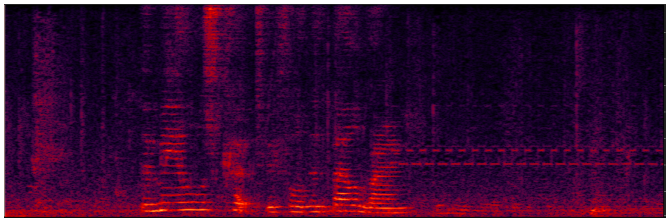


(a) MoDE architecture.

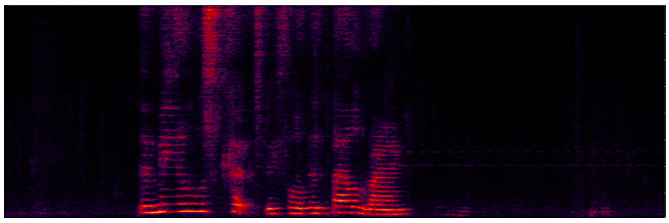


(b) Unsupervised clustering.

Figure 2.1: The MoDE algorithm and its initialization.



(a) Noisy.



(b) Enhanced.

Figure 2.2: Sonograms of noisy and enhanced speech in chatter noise.

the speech in the log-spectrum domain. We used clustering based on training of an autoencoder followed by k-means clustering in the embedded space. The clustering method is schematically depicted in Fig. 2.1b. As the number of experts m is not known in advance, it should be empirically determined. An example of the enhancement capabilities of the MoDE algorithm is depicted in Fig. 2.2.

In our ROS implementation, we used three experts. The total number of weights is 26,728,258. The algorithm can either run on a CPU or a GPU. The algorithm receives an audio frame size of 512 samples from ARI's microphone array (raw data of microphone #1) and provides an enhanced audio frame of the same size. The processed frame is published as 'enh_data' to the 'audio/enh_audio' topic. We can measure the timing of the package using a ROS subscriber ensuring we publish messages at 32.5Hz (32.5X512 = 16000 which is the sampling rate of the microphone array).

Although a significant noise reduction is obtained, as can be verified by both spectrogram assessment and informal listening tests, the performance of the ASR strongly depends on the aggressiveness of the algorithm (determined by the value of β).

2.2 ASR + MoDE integration performance

The enhanced speech is used by the ASR system to transcribe the utterance for further processing by the dialogue system. Currently, the dialogue system can only process English text, and hence French utterances should be transcribed to English.

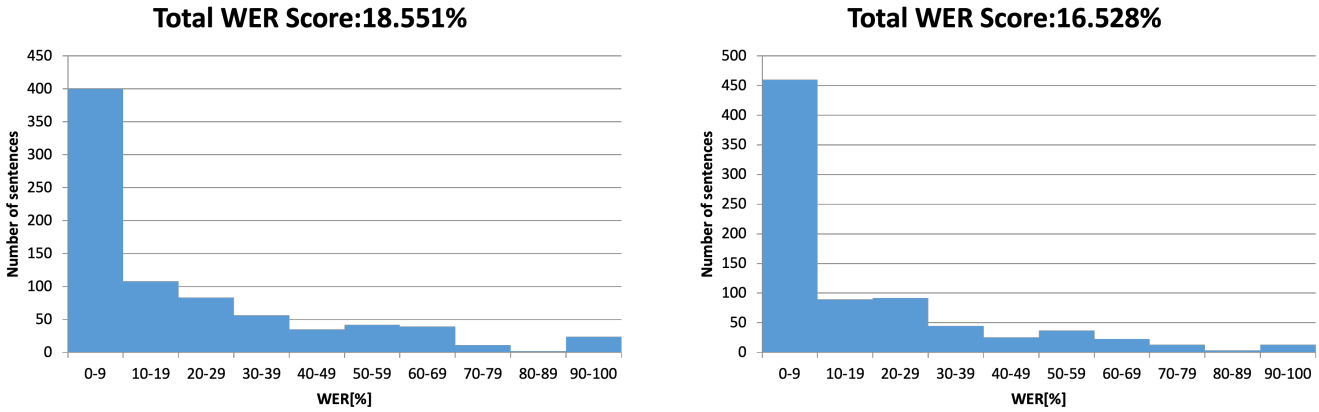
We tested and compared several ASR cloud services (Google, IBM, AWS, and Azure) and several on-premises services (Kaldi, Nvidia).

We started the evaluation using simulations. We convolved the entire test set of the TIMIT US English database [4] with the room impulse responses from the BIU database [5] (with three reverberation levels $T_{60} = 160, 360, 610$ ms) in three SNR levels 5, 10 and 15 dB. The results (not reported here) demonstrates the performance advantages of Google cloud service.

We continued by examining a small recording campaign carried out at Broca (the microphone was close to the speakers, hence reverberation was not an issue). In this dataset, we only have eight speakers, each uttering 100 French sentences. The word error rate (WER) of several ASR services can be found in Table 2.1. It is clearly demonstrated the Google and Microsoft Azure cloud services outperform the other solutions. We further compared the histogram of the WERs of Google cloud service and Nvidia's on-prem solution (in the task of French to French transcription).

Speaker	Kaldi	Google	AWS	IBM	Azure
1	0.26	0.14	0.28	0.27	0.12
2	0.21	0.14	0.33	0.28	0.16
3	0.29	0.15	0.36	0.28	0.12
4	0.30	0.15	0.34	0.29	0.18
5	0.29	0.17	0.43	0.34	0.15
6	0.36	0.19	0.47	0.37	0.15
7	0.32	0.17	0.51	0.35	0.30
8	0.26	0.13	0.41	0.34	0.20

Table 2.1: WER of several ASR services.



(a) Quartznet, Nemo platform,Nvidia.

(b) Google cloud service.

Figure 2.3: Histogram of WER.

Examination of the results in Fig. 2.3 reveals that both engines provide comparable results with slight advantage to the Google solution.

Finally, we examined the MoDE+ASR pipeline. We were able to show that by carefully tuning the noise reduction level (the frequency-dependent parameter β), we can find a working point for which the ASR performance improves, especially at the beginning of the utterance, when the ASR most struggles, probably since noise statistics is yet unavailable at the beginning of the utterance. We carried out a preliminary study at BIU acoustic lab, with reverberation level set to $T_{60} = 650$ ms and babble noise at SNR of approximately 5-10 dB. Results indicate the benefits of processing the microphone signal with MoDE prior to the application of the ASR engine (Google cloud solution).

2.3 Next steps

The performance of the algorithm with the data from Broca will be evaluated shortly. We will also continue the evaluation of on-prem solutions. Specifically, we are currently working on Nvidia's new ASR engine, RIVA. RIVA only provides a two-step solution, namely ASR from French utterance to French text followed by text translation from French to English, to be further processed by the dialogue system.[1] Such two-steps solutions are prone to accumulated errors. Another alternative is Google Anthos, that is claimed to provide comparable results to Google cloud service. Anthos also only provides a two-step solution. To mitigate this, we are currently translating the dialogue system to French and will conduct internal tests.

Currently, the enhancement-ASR-dialogue system can work in noisy environment with non-concurrent speaker. To allow more complex interactions, we will next incorporate our diariztion and separation algorithms. See discussion on the multi-party dialogue system in Sec. 3 below.

[1]The official French ASR will only be released on Q3 of this year. Currently, we try to work with Jarvis models, which is the previous version of the engine, imported to RIVA environment.

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement No. 871245.

3 Multi-party Dialogue Data Collection

To build the conversational system described in WP5 using machine learning approaches, we require realistic data. We, therefore, describe and motivate a data collection design for eliciting complex and natural multi-party conversations with a Social Robot. Dyadic data collections between single humans and robots focus on utterances directed at robots, but for multi-party conversation we also need observations of humans speaking to each other. Our design, therefore, focuses on eliciting conversation between all participants, and a particular concern is to generate data in which participants may have different goals and information. Here, acted role-play interactions are useful. These are often scripted and so can yield unrealistic data, so instead, our design uses pictograms for role-play task stimuli, leading to more realistic and spontaneous multi-party dialogue phenomena. Using this design, we have collected multi-party data with the ARI humanoid robot in the Broca Hospital as part of WP1 data collection tasks.

3.1 Design

The context for our data collection is a hospital scenario, where the robot plays the role of receptionist/helper. The system can answer FAQs regarding, e.g., patient schedules and catering facilities, and give directions to key hospital locations/facilities. The data collection is designed to capture 3-way conversations. Accordingly, we recruit pairs of participants each of whom are assigned to a role, either *patient*, or the patient's *companion*, a reasonable assumption based on observations made in WP1. The interactions themselves are split into 4 conditions with 4 task sets each and are designed to create realistic interactions between all parties by giving the participants goals related to the scenario (see Section 3.2).

To simulate natural human behaviour, when giving instructions to participants, it is important to not bias them to achieving the task in any specific way but to let them individually decide how to approach it. This is especially important for dialogue because there is always a danger of written instructions simply being read out loud to the robot. The work of [6] showed that meaning representations based on pictures can elicit more informative, more natural, more diverse, and better-phrased data, without priming participants to produce specific lexical items or phrases. Therefore, we use a set of pictograms, given to participants in the task instructions sheet, to provide them with goal representations for their tasks. Figure 3.1 shows examples of the picture representations.

3.2 Implementation

3.2.1 Physical Sensor Setup

The data collection is carried out with the ARI humanoid robot capturing and record the audio and video of the whole interaction from its perspective. Additionally an external camera is used to record from a third-party perspective.

The robot is stationary throughout. Verbal interaction with users is controlled by a Wizard of Oz (WoZ) operator, who observes and listens to the conversation remotely, and whenever the robot is addressed selects an appropriate response from a menu on-screen. The robot then verbalises the response using Acapela Text-To-Speech[1]. For the design used here, we had a total of 17 responses the wizard could select from. This includes answers to questions that will arise from the tasks and general greetings. In addition, some social responses can be added such as yes/no, please/thank you, and "I don't know". With this low number of possible responses, the wizard is able to select the most appropriate one without delay.

3.2.2 Interaction Scenario

For this data collection, participants will assume either of two roles, as designated by the experimenter:

[1] https://www.acapela-group.com/

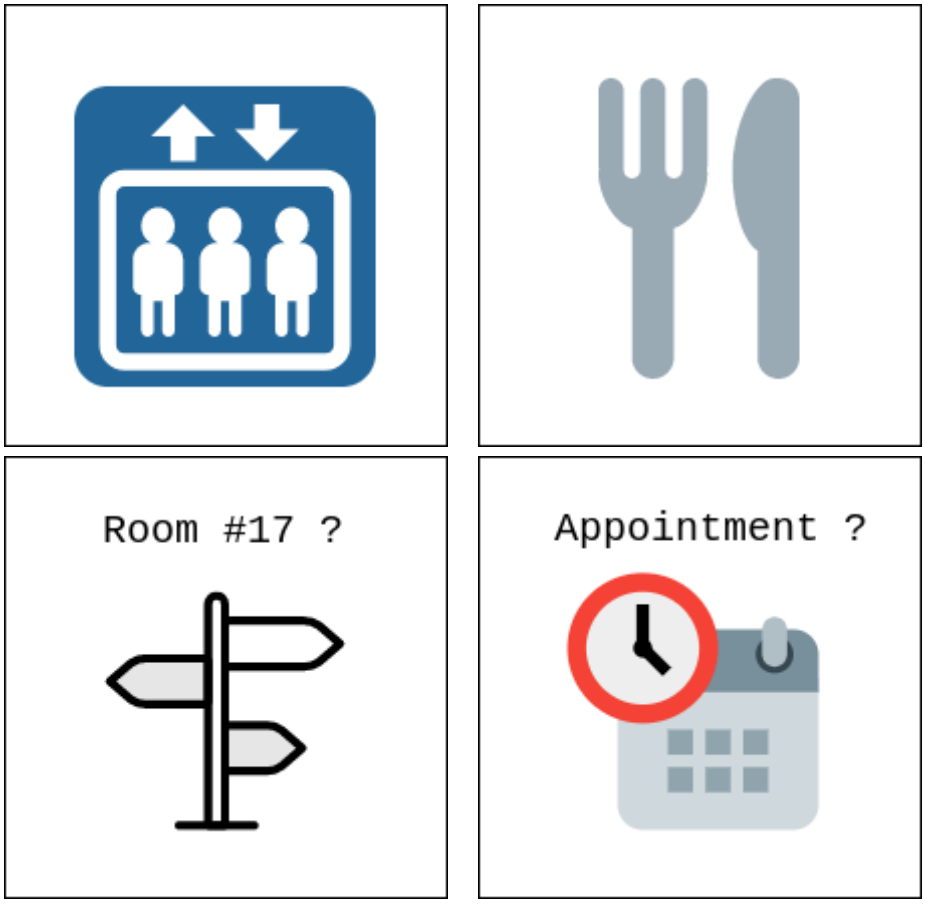


Figure 3.1: Sample picture representations of task goals given to participants (find elevator / cafe / room 17 / appointment time)

- ***Patients:*** *are visiting a daycare hospital to attend appointments with different healthcare professionals. They have brought a friend or family member to help them and keep them company.*
- ***Companions:*** *are accompanying a friend or family member on their visit to the daycare hospital. They have volunteered to do so in order to help them and keep them company.*

Participants will enter the 'reception room' at a hospital[2], and will need to find information about certain things, locations, or events. Each participant will be given an information sheet for each part of the data collection that will provide them with a role description and images representing the information that they are looking for. In the reception room, they will encounter the robot that is there to help.

The general instructions on the information sheet will inform participants of their overall goals, but they'll be given freedom to interact with each other and the robot as they see fit to accomplish them.

3.2.3 Conditions and Tasks

The interactions themselves are split into 4 conditions (A–D) with 4 task sets (1–4) in each condition. Each pair of participants should perform all the task sets in all the conditions. This makes for a total of 16 interactions. For each pair of participants, the order in which they enact the conditions should be randomised.

Condition A: helpful companion

In this condition the *patient* role is instructed to talk to the robot and their companion to try to find the information that their goals require. They have 2 goals, given to them in the instruction sheets via picture representations (see Figure 3.1). The goals are different in each of the 4 task sets. The *companion* role is not given any explicit informational goals, but they are instructed that they are there to support the *patient* and keep them company throughout their day, and that the robot in the reception room might be able to help find the answers the *patient* is looking for. This is the same for all task sets in this condition.

In this way we elicit multi-party conversations where one human (companion) attempts to find out the goals of another human (patient), while the patient can also directly address the robot.

Condition B: shared goals

In this condition, the *patient* and the *companion* role are each given the same shared goals in their task instruction sheets for each of the 4 task sets. The task in this condition follow the same goals as the *patient* task instruction sheets for condition A.

Condition C: reluctant patient

This condition is similar to ***Condition A***, the *patient* role has the same 4 task sets with the same goals as in the previous conditions. But now they are instructed that they do not want to talk with the robot. This is done to try to collect data in situations where one of the participants is reluctant to interact with robots. It could also model cases of social anxiety where a human is not keen to talk to a receptionist, for example.

The *companion* role is given the same task instructions as in ***Condition A***, with no explicit goal instructions except to help the *patient* find the answers they are looking for. This remains the same for all task sets in this condition. In this way we elicit conversations where a robot can listen to humans providing goals to each other. For example the reluctant patient will be more likely to express their goals to the companion rather than directly to the robot.

Condition D: different goals and info

For this condition, the *patient* and the *companion* role are each given 1 goal, but their goals are different from each other in each of the 4 task sets. In addition, the *patient* role is provided with the information which the *companion* needs to complete their goal (for example the location of the cafe, when the companion wants a drink).

This models situations where dialogue participants have different information, some of which is needed to complete their dialogue partner's goals. For example, the companion may want to get a coffee, and the patient knows where the cafe is (e.g. from a previous visit to the hospital). In this way we elicit conversations between the 2 humans which a robot should be able to listen to and understand – for example tracking the different goals of the 2 humans and the fact that a goal has been met (e.g. if the patient tells the companion where the cafe is located).

[2] A mock up at the Broca living lab.

Patient: I'm thirsty
Companion: Do you want me to get you something to drink?
Patient: Yes, I think there is a cafe
Companion: Do you remember where it is?
Patient: No sorry
Companion: I will go ask where
Patient: Okay
Companion: [addressing the robot] Where is the cafe?
Robot: [Gives directions to cafe]

Table 3.1: Example Dialogue - Condition C

Patient: [addressing robot] Where's room 17 please?
Companion: [addressing patient] I need to go get a coffee first.
Patient: [addressing companion] Oh it's upstairs.
Companion: [addressing patient] OK fine, just wait for me then.
Robot: [Gives directions to room 17]

Table 3.2: Example Dialogue - Condition D

3.2.4 Collected data

We have conducted a first data collection at the Broca day-care hospital where we invited volunteers, including elderly people, to interact with the robot, after briefing from Broca's living lab staff. We have so far recorded 21 individuals over 3 days using this design. Not all conditions could be recorded as the tasks were too complex to understand for some of the patients.

3.3 Next Steps

Future work will explore more complex multi-party situations. One example is where 2 participants have *conflicting* information that must be resolved (for example they might disagree about the location of the next appointment). In this case the robot should track the conflicting information and give the information that it has (e.g. the location of the next appointment) to resolve disagreements.

Our immediate next goal is to use the collected data to develop a system that is able to track multiple people in conversation. An "Intelligent Listener" dialogue system should ultimately try to track what each speaker is saying, what their individual goals/questions are, etc. As well as correctly tracking the goals of each user it should be able to determine which goals have been met and which still need to be addressed. It could also try to determine points in the human-human conversation at which it is possible or desirable to "barge-in" with information meeting a user's goals.

At the time of writing this deliverable, the data collected in the Broca living lab is still being translated by professional translators employed by HWU. Once this translation has finished, we will assess the data and the need for additional data collection locally.

This project has received funding from the European Union's Horizon 2020
Research and Innovation Programme under Grant Agreement No. 871245.

4 Outputs

Various software modules for ASR, speech audio enhancement algorithms, and the conversational system, have been implemented and integrated with the ARI robot system.

The software for the ASR is developed in [10], provides a ROS wrapper for Google's speech cloud services, for the following google cloud API: Speech-to-Text [1]; Media Translation [2]; and Cloud Translation [3]. The MoDE algorithm has been implemented in a ROS package, and can be found in [9]. The dialogue system software is being updated in [11]. The software packages forming the conversational system have been dockerized and are grouped and available in the repository found in [11].

We stress that, at this stage, the software package (with all its ingredients, namely noise reduction, ASR, and conversational system) only supports a single speaker. Hence, concurrent speakers are not supported. We are currently finalizing a speaker extraction/separation algorithm, and the associated software package, that will provide several audio streams, one for each relevant speaker in the system. Each audio stream will be processed by an instance of an ASR engine. The conversational system will then process the separated streams and analyze the entire multi-party interaction with the robot. To this end, the NLU, NLG, and Dialogue Management of the conversational system will be retrained/reprogrammed to work with multiple input streams using the output of WP3 and the data collection described above.

As per European Commission requirements, the repositories will be available to the public for a duration of at least four years after the end of the SPRING project. People can request access to the software to the project coordinator at `spring-coord@inria.fr`. The software packages use ROS (Robotics Operating System) to communicate with each other and with the modules developed in the other workpackages.

[1] `https://cloud.google.com/speech-to-text`
[2] `https://cloud.google.com/media-translation`
[3] `https://cloud.google.com/translate`

Bibliography

[1] Shlomo E. Chazan, Sharon Gannot, and Jacob Goldberger. A phoneme-based pre-training approach for deep neural network with application to speech enhancement. In *International Workshop on Acoustic Signal Enhancement (IWAENC)*, Xi'an, China, September 2016. Best paper award.

[2] Shlomo E. Chazan, Jacob Goldberger, and Sharon Gannot. Speech enhancement with mixture of deep experts with clean clustering pre-training. In *IEEE International Conference on Audio and Acoustic Signal Processing (ICASSP)*, Toronto, Ontario, Canada, June 2021.

[3] Ori Ernst, Shlomo E. Chazan, Sharon Gannot, and Jacob Goldberger. Speech dereverberation using fully convolutional networks. In *The 26th European Signal Processing Conference (EUSIPCO)*, Rome, Italy, September 2018.

[4] John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, and David S Pallett. Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1. *NASA STI/Recon technical report n*, 93:27403, 1993.

[5] Elior Hadad, Florian Heese, Peter Vary, and Sharon Gannot. Multichannel audio database in various acoustic environments. In *International Workshop on Acoustic Signal Enhancement 2014 (IWAENC 2014)*, Antibes - Juan les Pins, France, September 2014.

[6] Jekaterina Novikova, Oliver Lemon, and Verena Rieser. Crowd-sourcing NLG data: Pictures elicit better data. In *Proceedings of the 9th International Natural Language Generation conference*, pages 265–273, Edinburgh, UK, September 5-8 2016. Association for Computational Linguistics.

[7] SPRING Project. D5.1: Initial high-level task planner and conversational system prototype for realistic environments. `https://spring-h2020.eu/wp-content/uploads/2021/06/SPRING_D5.1_Initial_High-level_Task_Planner_and_Conversational_System_Prototype_for_Realistic_Environments_vFinal_31.05.2021.pdf`.

[8] SPRING Project. D6.2: Specifications of the generator of situated interactions. `https://spring-h2020.eu/wp-content/uploads/2021/06/SPRING_D6.2_Specifications-of-the-generator-of-situated-interactions_VFinal_31.05.2021.pdf`.

[9] SPRING-WP3-Repository. SPRING-WP3-Repository-MoDE. `https://gitlab.inria.fr/spring/wp3_av_perception/speech-enhancement/-/tree/main`.

[10] SPRING-WP5-Repository. SPRING-WP5-Repository-ASR. `https://gitlab.inria.fr/spring/wp5_spoken_conversations/asr/-/tree/pre-integration-1`.

[11] SPRING-WP5-Repository. SPRING-WP5-Repository-Docker. `https://gitlab.inria.fr/spring/wp5_spoken_conversations/docker-wp5-spoken-conversations/-/tree/pre-integration-1`.

[12] Yochai Yemini, Ethan Fetaya, Haggai Maron, and Sharon Gannot. Scene-agnostic multi-microphone speech dereverberation. In *Interspeech*, Brno, The Czech Republic, 2021.