



Deliverable D5.4: Multi-party conversational system in target environments

Due Date: 30/11/2023

Main Author: HWU

Contributors: HWU, BIU, INRIA

Dissemination: Public Deliverable

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



DOCUMENT FACTSHEET

Deliverable	D5.4: Multi-party conversational system in target environments
Responsible Partner	HWU
Work Package	WP5: Multi-User Spoken Conversations with Robots
Task	T5.3: Multi-party Conversational System
Version & Date	30/11/2023
Dissemination	Public Deliverable

CONTRIBUTORS AND HISTORY

Version	Editor	Date	Change Log
0.1	HWU	16/10/2023	Initial Draft
1.0	HWU	30/10/2023	First Draft
1.1	BIU	14/11/2023	Second Draft
1.2	HWU	23/11/2023	Third Draft
1.3	BIU	26/11/2023	Fourth Draft
2.0	HWU	30/11/2023	Final Draft

APPROVALS

Authors/editors	HWU
Task Leader	HWU
WP Leader	HWU



Contents

Abbreviations	3
Executive Summary	4
1 Introduction	6
2 Multi-Party Data Collection	7
2.1 Design	7
2.2 Collection	7
2.3 Annotations	8
3 Multi-Party ASR	9
3.1 Audio Pipeline	9
3.1.1 Noise Reduction	10
3.1.2 Speaker Extraction	10
3.1.3 Speaker Separation	10
3.1.4 Speaker ID	11
3.1.5 NVIDIA Riva ASR	12
3.2 Audio Data Collection	12
3.2.1 Dataset of French Utterances	12
3.2.2 ASR Accuracy	12
3.2.3 Acoustic Setup	13
3.2.4 Assessment of the Noise Reduction Algorithm	13
3.2.5 Assessment of the Speaker Separation Algorithm	14
4 Multi-Party Conversational System	15
4.1 Initial Modular Architecture	15
4.2 LLM-based Architecture	17
4.2.1 Addressee Detection	18
4.2.2 Partial Utterance Detection and Clarification	19
4.2.3 Hospital Information Response	20
4.3 Multi-Party Example Interaction	21
4.4 Performance of the Conversational System	21
5 Implementation	22
5.1 Outputs	22
5.2 LLM Server	22
Bibliography	23

Abbreviations

Abbreviation	Meaning
AI	Artificial Intelligence
AP-HP	Assistance Publique – Hôpitaux de Paris (SPRING Partner)
ARI	Social assistive robot used by the SPRING project
ASR	Automatic Speech Recognition
BIU	Bar-Ilan University (SPRING Partner)
CA	Conversational Agent
CS	Conversational System
DM	Dialogue Management
CVUT	Czech technical university in Prague (SPRING Partner)
ERM	ERM Automatismes Industriels (SPRING Partner)
FAQ	Frequently Asked Questions
HRI	Human Robot Interaction
HWU	Heriot-Watt University (SPRING Partner)
INRIA	Institut National de Recherche en sciences et technologies du numérique (SPRING Partner)
LLM	Large Language Model
MPC	Multi Party Conversation
NLG	Natural Language Generation
NLP	Natural Language Processing
NLU	Natural Language Understanding
PAL	PAL Robotics (SPRING Partner)
PNP	Petri-Net Planner
RAG	Retrieval Augmented Generation
ROS	Robot Operating System
SPRING	Socially Pertinent Robots in Gerontological Healthcare
UNITN	University of Trento (SPRING Partner)
WP	Work Package (of the SPRING project)

Executive Summary

Deliverable D5.4 reports on the final software components for multi-party interaction including the conversational system, NLU, dialogue management, and NLG components geared towards dealing with realistic scenarios.

The work reported in this deliverable was carried out to fulfil the objectives of task 5.3, in this deliverable we present the development of the multi-party conversational system.

1 Introduction

The overall objective of WP5 (Multi-User Spoken Conversations with Robots) is to develop techniques for multi-user conversation involving a robot and multiple humans. This objective will enable sensor-based (data-driven) and knowledge-based robot actions for multi-modal multi-person interaction and communication.

This deliverable reports on the final software components for the multi-party interaction including the ASR, the conversational system, NLU, dialogue management, and NLG components geared towards dealing with realistic scenarios.

This document describes the development of the SPRING multi-party conversational system. Section 2 describes the data collection efforts for eliciting complex and natural multi-party conversations with a Social Robot for training the multi-party conversation system. In Section 3 we describe the multi-party automatic speech recognition (ASR) module. In Section 4 we describe the conversation system for multi-party interactions. Section 5 describes the final implementation and outputs of the Multi-party Conversational System.

The software will be released in the code repositories for WP5 [45], WP3 [43], and WP4 [44].

As per European Commission requirements, the repository will be available to the public for at least four years after the end of the SPRING project. People can request access to the software from the project coordinator at spring-coord@inria.fr. The software packages use ROS (Robotics Operating System) [46] to communicate with each other as well as the modules developed in the other work packages.

2 Multi-Party Data Collection

We require high-quality and relevant corpora to be able to use data-driven models to tackle multi-party dialogue challenges for the SPRING project.

A comprehensive summary of existing multi-party corpora revealed that they are all human-human datasets [28]. Therefore, for SPRING we conducted a data collection where a social robot is also one of the interlocutors. This is an important distinction as there are differences in behaviour when talking to a robot compared to a human [35, 24]. Collecting multi-modal data of real human-robot interactions was therefore critical to create a multi-party social robot, as will be the case for the ARI robot in SPRING.

2.1 Design

The context for our data collection is the hospital scenario of SPRING, where the robot plays the role of receptionist/helper. The system can answer FAQs about topics like patient schedules, catering facilities, directions to key hospital locations, etc. The data collection is designed to capture 3-way conversations with pairs of participants and a robot. Each participant is then assigned one of two roles: the *patient*, or the patient's *companion* – a reasonable assumption in this context. An information sheet is provided with their role description and an image representing their goal. Participants will enter the 'reception room' at Broca hospital and will encounter the robot helper. They are given complete freedom to interact with each other and the robot as they see fit to accomplish their goals.

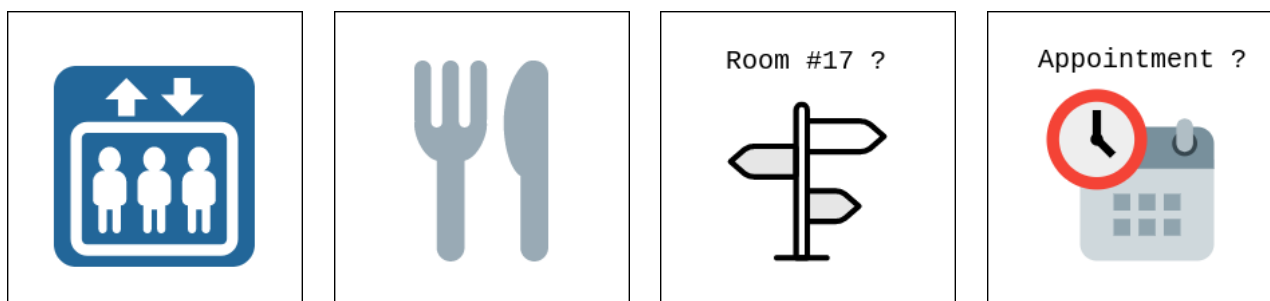


Figure 2.1: A sample of the pictograms used to represent user goals, given to patients and companions (find elevator/cafe/room 17/appointment time). These elicited dialogues without restricting vocabulary.

We give participants pictograms in the task instructions sheet to provide them with their goal, Figure 2.1 shows examples of the pictograms, in order to not bias how they individually approach achieving their goal. 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 [30].

Full details from the design of the data collection for multi-party task-based dialogue can be found in [4].

2.2 Collection

Data Collection was carried out by AP-HP researchers. The experiments were held in the Living Lab of the Broca Hospital (AP-HP) where volunteers were recruited to participate in the experiments. A description of the Day Care Hospital environment, recruitment, and methodology for working with participants can be found in deliverables D1.4 [39] and D1.5 [40].

At the time of writing a total of 91 interactions (both single and multi-party) have been collected with participants in the hospital. Table 2.1 summarises the different phases of data collection.

Table 2.1: Multi-Party Data Collection Phases.

Dates	Location	System	Num Interactions
April 2022	Broca Living Lab	Wizard of Oz	35
17th-21st October 2022	Broca Living Lab	Integrated V1	10
29th-31st March 2023	Broca Living Lab	Multi-Party V1	10
11th May-18th September 2023	Day-care Hospital	Integrated V2	15
19th Sep-22nd Nov 2023	Day-care Hospital	LLM V1	21

2.3 Annotations

We have, so far, transcribed, translated (from French to English), anonymised, and annotated 29 multi-party conversations (774 turns) from the first experiment wave in April 2022.

An annotated extract from an MPC in our collected corpus can be found in Table 2.2. In turn 1, U1 states that “we’d like a coffee”, indicating that U1 and their companion U2 would *both* like a coffee. This turn is annotated with two intents: *greet* (due to the “hello”), and *request*. This request intent has a slot value to indicate that the request is for a beverage – coffee. The goal tracking annotation signifies that a goal has been established in this turn with ‘G’. The goal is shared by ‘U1+U2’, and their goal is to drink a coffee. In turn 2, ARI responds informing both users where the café is, hence the *inform* intent annotation. The goal tracking annotation is the same as turn 1, but starts with ‘AG’ (for ‘answer-goal’) instead of simply ‘G’. This indicates that this goal has been answered, which is critical knowledge for the system to track which goals remain open. In this example, the goal is explicitly closed in turn 3, indicated by the corresponding ‘CG’ (close-goal) goal-tracking annotation. Not all goals are explicitly closed by the user. A dialogue manager could decide to implicitly close an answered goal if the user does not reopen it within three turns, for example. We only annotate explicit goal closures, like the one in turn 3. There are two intents annotated in both turns 1 and 3 in Table 2.2, and multiple goal annotations can similarly exist, separated by a semicolon. For example, “I’m hungry but need the toilet first” simultaneously opens two goals. All of these annotations were completed using the ELAN tool [7].

Table 2.2: A corpus example displaying shared goals with both intent-slot and goal-tracking annotations.

	User	Utterance	Intent-Slot Annotation	Goal Tracking Annotation
1	U1:	Hello, we’d like a coffee. Where can we go?	<i>greet()</i> ; <i>request(drink(coffee))</i>	G(U1+U2, drink(coffee))
2	ARI:	You have to enter the building behind you.	<i>inform(directions(cafe))</i>	AG(U1+U2, drink(coffee))
3	U2:	Ok, well thank you very much.	<i>acknowledge()</i> ; <i>thank()</i>	CG(U1+U2, drink(coffee))

This annotated data has been used to continuously evaluate the performance of the conversational system developed in Chapter 4.

3 Multi-Party ASR

3.1 Audio Pipeline

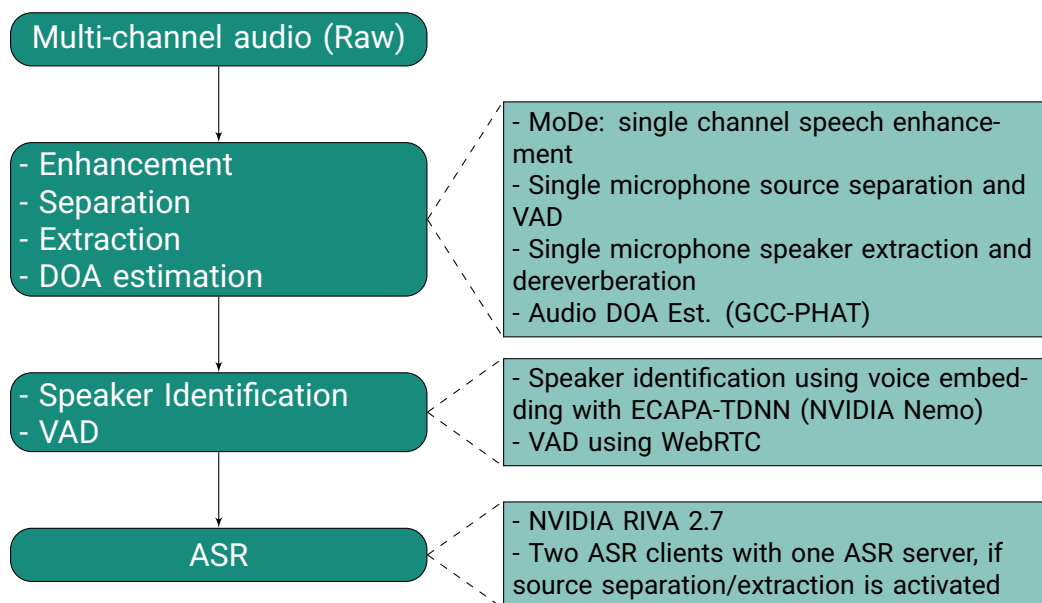


Figure 3.1: Simplified Audio Pipeline of the Multi-party ASR

In real-life scenarios, a robot may engage with a group of people amid noisy and reverberant surroundings. The audio pipeline's objective is to generate multiple streams of transcribed speech from all speakers, which the conversational system will utilize. The transcribed text streams should maintain consistency over time. This may require various methods of attributing identity to the speakers, including direction of arrival (DOA) estimation and speaker identification.

We limit the scenarios to a maximum of two concurrent speakers. We also assume that the robot interacts with individuals in a half-duplex manner; namely, it does not listen while talking.

The audio pipeline encompasses a four-step process as outlined below. It can be viewed schematically in Fig. 3.1.

1. Acquire raw audio from the ReSpeaker microphone array.
2. Enhance the speech quality through various algorithms, including noise reduction, source separation, and speaker extraction. Additionally, the DOAs of the speakers in the scene is estimated.
3. Apply voice activity detector (VAD). When speech is active, a speaker identification module is activated to provide speaker embedding, solely characterizing the speaker's voice.
4. Transcribe the enhanced audio data using the RIVA automatic speech recognition (ASR) engine. The resulting streams of transcribed text (maximum two) are fed into the dialogue manager for further processing.

This process was designed to comprehensively handle the intricacies associated with understanding both single and multiple speakers, using a systematic approach for audio processing and analysis.

3.1.1 Noise Reduction

The noise reduction module [8] presents a single-microphone solution founded on a Mixture of Deep Experts (MoDE) neural-network architecture. This “Divide and Enhance” structure encompasses a collection of deep neural network (DNN)-based sub-networks, each acting as an ‘expert’ attuned to a distinct speech spectral pattern, such as a phoneme. Each expert generates a speech presence probability (SPP) map, determining whether a time-frequency bin is predominantly speech or noise based on its expertise. The final time-frequency mask is derived by weighting the SPP estimates from various experts, using weights determined by the gating DNN. Subsequently, a soft spectral attenuation, guided by the estimated mask, is applied to enhance the noisy speech signal. The experts’ specialization enhances robustness to unfamiliar noise types and concurrently reduces inference complexity. A block diagram illustrating the noise reduction algorithm is presented in Fig. 3.2.

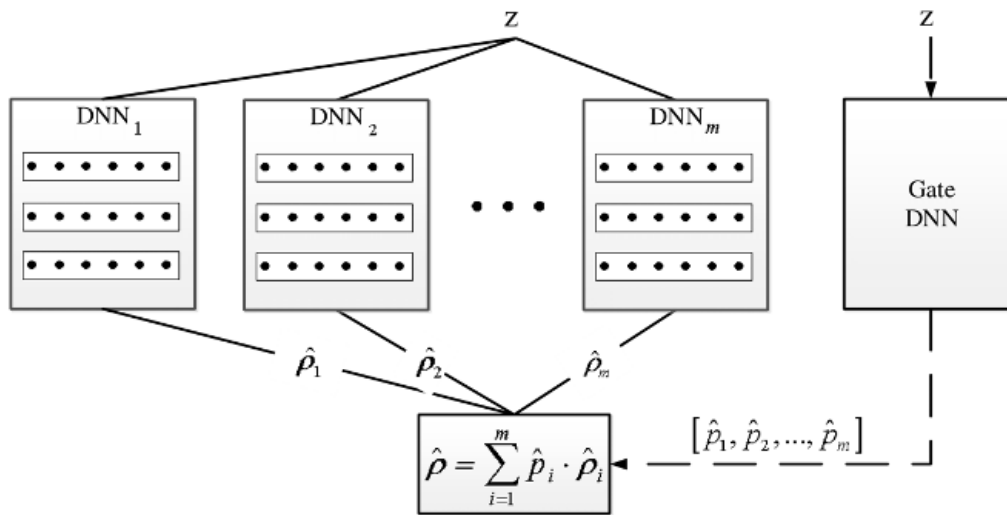


Figure 3.2: Block diagram of the mixture of deep experts (MoDE) algorithm.

3.1.2 Speaker Extraction

The speaker extraction module is underpinned by a two-stage method [17]. This architecture consists of two sub-networks: one dedicated to the extraction task and the other to dereverberation and residual interference suppression. Beginning with a reference signal of the desired speaker, the module initially extracts the clean yet still reverberant speaker from the noisy-mixed signal. In the subsequent stage, the extracted signal undergoes further enhancement through joint dereverberation, coupled with residual noise and interference reduction. The method’s block diagram is illustrated in Fig. 3.3. Through simulation studies, it has been demonstrated that this extraction algorithm surpasses state-of-the-art (SOTA) algorithms when applied to speech mixtures in realistic and adverse acoustic conditions. A noteworthy feature of the speaker extraction algorithm, particularly pertinent to the SPRING project, is its capability to estimate speaker embeddings that can be leveraged for speaker identification tasks.

3.1.3 Speaker Separation

The speech separation module implements a recent single-microphone speaker separation algorithm [31] aiming at noisy and reverberant environments typical to real-world environments. Two variants of the algorithm are available, Separation TF Attention Network (Sep-TFAnet) and Sep-TFAnet^{VAD}. The latter incorporates a VAD into the separation network. The separation module is based on a temporal convolutional network (TCN) backbone inspired by the Conv-Tasnet architecture with multiple modifications. Rather than a learned encoder and decoder, we use short-time Fourier transform (STFT) and inverse short-time Fourier transform (iSTFT) for the analysis and synthesis, respectively. Since the system is specially developed for human-robotic interactions, it is designed to support online mode. The Sep-TFAnet^{VAD} variant provides activity patterns of the separated speakers and can serve for diarization in the downstream dialogue manager. The separation capabilities of Sep-TFAnet^{VAD} and Sep-TFAnet were evaluated using simulated data under several acoustic conditions, demonstrating their advantages over competing methods. In this document, we

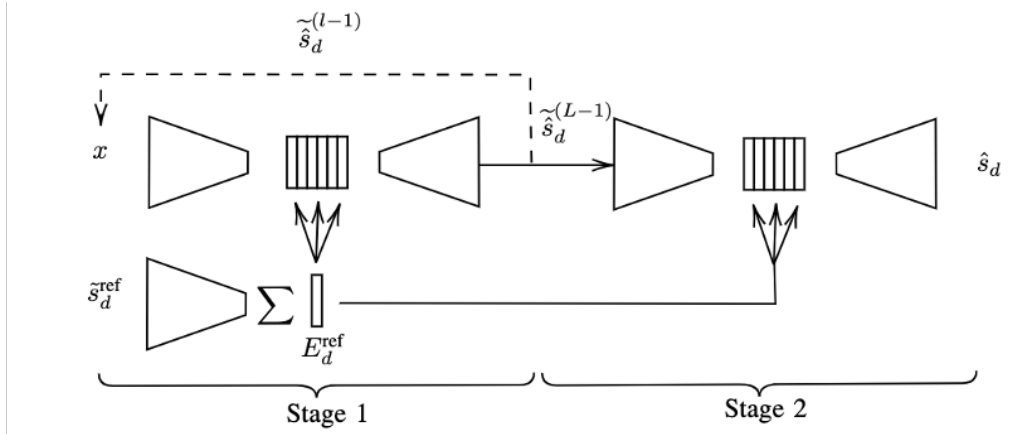


Figure 3.3: Speaker extraction using a reference signal.

will demonstrate the separation results of data obtained by ARI.¹ The block diagram of the separation algorithm is illustrated in Fig.3.4.

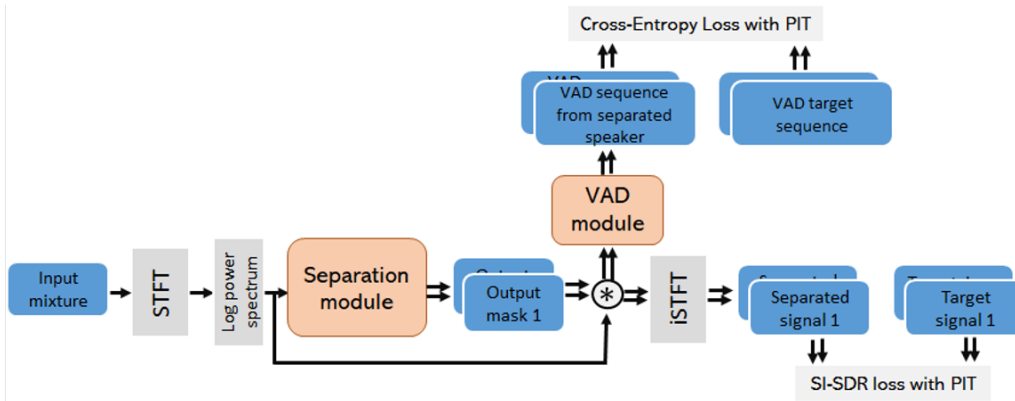


Figure 3.4: Separation algorithm.

3.1.4 Speaker ID

Various attributes influence the speaker's identity, differentiating between short-term and long-term identification. Short-term identification relies partly on spatial information derived from the current scene. This involves employing a late fusion mechanism that combines visual-based and audio-based DOA estimation.

We employed a time difference of arrival (TDOA) estimation for the audio modality, utilizing an instantaneous version of the generalized cross-correlation phase transform (GCC-PHAT). This implementation utilizes two horizontal microphones from the Respeaker microphone array embedded in ARI. The TDOA readings are then translated into DOA estimates using geometric considerations, and these estimates are further enriched with spatial information obtained through visual means. Additionally, depth information obtained through the visual modality proves valuable in accurately determining the DOA, particularly in scenarios where speakers are close to the robot. It is noteworthy that while the GCC-PHAT algorithm reliably provides TDOA readings in frames with a single speaker, a robust extension of the GCC-PHAT [37] will be explored for cases involving concurrent activity of multiple speakers, should an audio-based DOA estimate be necessary for such scenarios.

For long-term speaker identification, we depend on Nvidia's ECAPA-TDNN model to extract speaker embeddings, producing a 192-dimensional voice signature vector.² The speaker identification module stores embeddings linked to specific speakers in an internal database. A specific speaker is identified when the cosine similarity between its embedding and an entry in the database is higher than a predefined threshold.

Nevertheless, the existing method exhibits inconsistency in scenarios with multiple speakers, resulting in arbitrary speaker switches. To tackle this issue, we are presently implementing a more robust solution that integrates the

¹<https://Sep-TFAnet.github.io>

²https://catalog.ngc.nvidia.com/orgs/nvidia/teams/nemo/models/ecapa_tdnn

speaker embedding model with segmentation and clustering techniques for enhanced results [12]. Initial results are promising. Finally, the speaker embedding obtained from the speaker extraction algorithm can serve as a speaker ID, at least during the conversation.

3.1.5 NVIDIA Riva ASR

We have implemented Nvidia's on-premise RIVA version 2.7, which includes an updated language model and comprehensive support for the French language, encompassing punctuation and capitalization.³ To enhance sentence parsing accuracy, we integrated a DNN-based VAD and introduced a configurable pause time between sentences. This ensures improved handling and processing of spoken content to avoid incorrect sentence segmentation.

3.2 Audio Data Collection

The audio pipeline's performance underwent assessment through a series of experiments. The primary objective of this processing is to enhance ASR performance within the challenging hospital setting characterized by reverberation and noise. To simulate these conditions accurately, the acoustic laboratory at Bar-Ilan University (BIU) was employed.

3.2.1 Dataset of French Utterances

Through a collaboration between BIU and Harriot Watt University (HWU), a dataset was created that comprises common phrases typically encountered in a hospital setting. The HWU team curated 100 French sentences and assigned specific intents to each of them.

The recording campaign was held at the Broca Hospital and was organized by the hospital team. The sentences were uttered by eight individuals, categorized as two elderly men, two elderly women, two young men, and two young women, and captured using a close-talk microphone. To streamline the data collection process, BIU designed custom software with an accompanying user interface (refer to Fig. 3.5). Overall, 423 utterances were recorded.

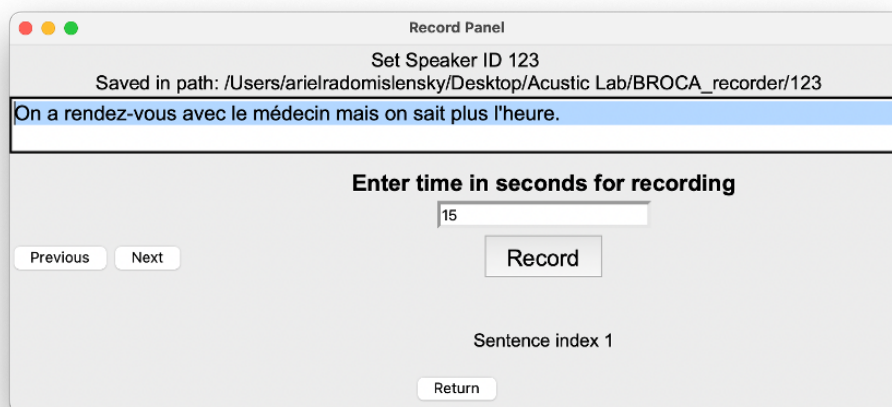


Figure 3.5: Screenshot from the user interface for the data collection at BROCA.

3.2.2 ASR Accuracy

Alongside this dataset, an alternative scoring system, the semantic error rate (SER), was introduced by HWU to complement the widely used word error rate (WER). While WER effectively measures overall system performance, SER introduces the critical dimension of semantically comprehending the intent of the sentence by the dialogue manager. Using both metrics provides a more comprehensive assessment of our audio pipeline.

³<https://docs.nvidia.com/deeplearning/riva/user-guide/docs/release-notes.html#riva-release-2-7-0>

3.2.3 Acoustic Setup

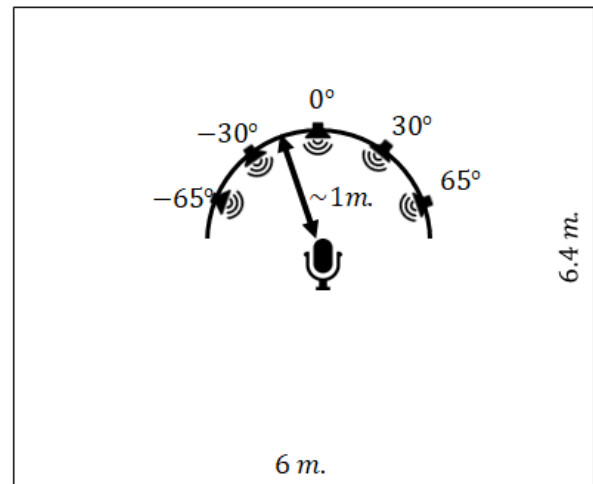
To analyze the algorithms' performance in a real-world setting, specifically in the context of robot audition, a complementary recording campaign was held at the acoustic laboratory at Bar-Ilan University. This lab is a $6 \times 6 \times 2.4$ m room with a reverberation time controlled by 60 interchangeable panels covering the room facets. In our experiments, the reverberation time was set (by changing the panel arrangements) to either 350 ms, typical of a meeting room, or 600 ms, typical of a lecture hall. The room layout is depicted in Fig. 3.6a.

ARI is equipped with a ReSpeaker 4-Microphone Array v2.0,⁴ installed inside the robot's compartment, 80 cm above its base. We only used one of the microphones for evaluating the noise reduction and the separation algorithms.

In our experimental setup, ARI was centrally positioned in the acoustic lab, surrounded by loudspeakers arranged in a semi-circle with a radius of approximately 1 m. Five loudspeakers were positioned on the perimeter of the semi-circle at $[-65, -30, 0, 30, 65]^\circ$. The experimental setup is schematically depicted in Fig. 3.6b.



(a) Lab view with ARI and loudspeakers.



(b) Schematic view of the experimental setup.

Figure 3.6: Experimental setup at BIU acoustic lab.

3.2.4 Assessment of the Noise Reduction Algorithm

To assess the efficacy of the MoDE algorithm [8], we played the 423 utterances recorded at Broca through randomly selected loudspeakers at the two tested reverberation levels. Additionally, babble noise, generated by transmitting multiple speakers from eight surrounding loudspeakers, was separately recorded. The clean and reverberant speech signals were combined with the noise signal at a signal-to-noise ratio (SNR) level of 12 dB. In Fig 3.7, we depict a histogram illustrating the occurrences of each WER level for (a) the clean and reverberant signals, (b) the noisy signals, and (c) the enhanced signals. MoDE demonstrates the capability to reduce the WER.

⁴https://wiki.seeedstudio.com/ReSpeaker_Mic_Array_v2.0/

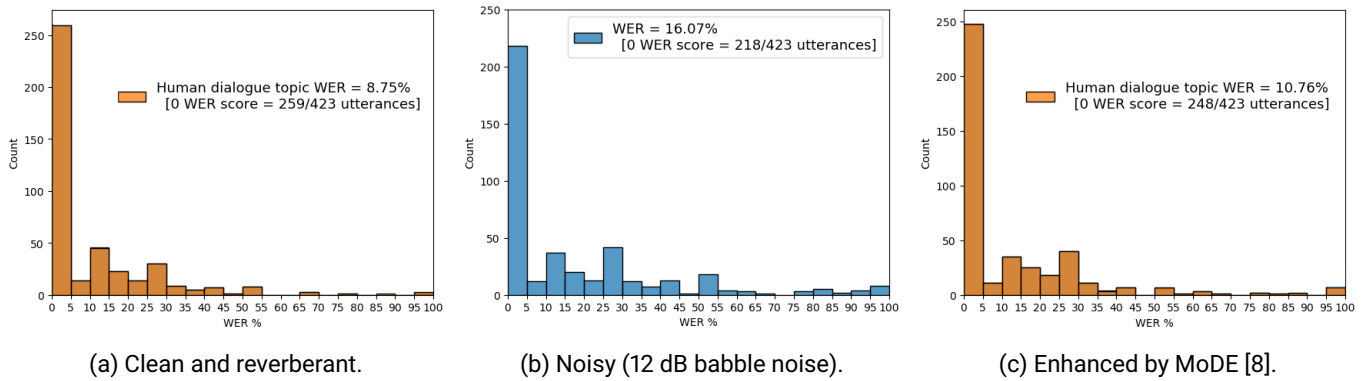


Figure 3.7: ASR histograms for $T_{60} = 350$ ms. Average WER and SER scores are also depicted.

3.2.5 Assessment of the Speaker Separation Algorithm

To assess the separation algorithm [31], we constructed an additional dataset comprising a mixture of speakers. Each sample was created by randomly selecting two loudspeakers (refer to Fig. 3.6b) and playing speech utterances randomly chosen from the Librispeech test set [32]. These utterances were individually recorded by ARI and manually mixed for subsequent Signal-to-Interference plus Noise Ratio (scale-invariant signal-to-distortion ratio (SI-SDR)) calculation. The overlap between speakers was randomly set within the range of [25%, 50%]. No external noise was introduced, so only sensor and low-level ambient noise were present in these recordings. In total, 200 samples were generated at each reverberation level.

The histograms in Fig 3.8 depict the frequency of occurrences per WER level. Significant performance improve-

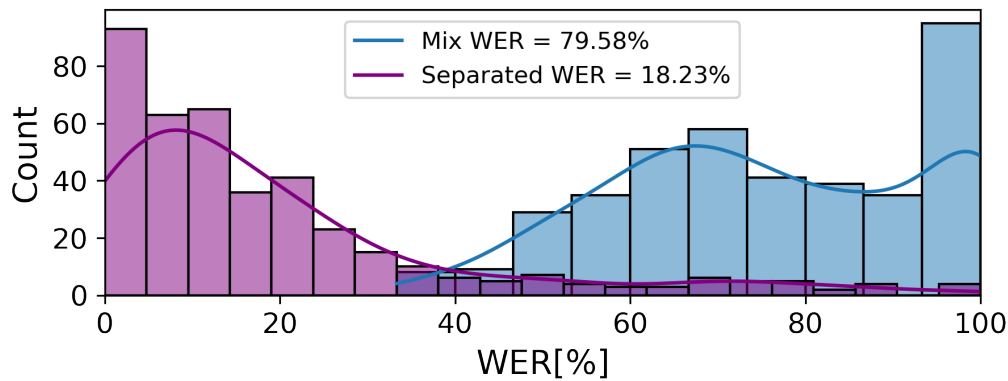


Figure 3.8: WER results with ARI recordings, $T_{60} = 0.35$ s and low sensor noise. The overlap of the speakers was randomly set to the range [25%, 50%].

ments in the WER are evident, with a notable improvement from 76% to 18.23% in the mean WER. Additionally, a distinct shift in the histogram towards lower WER levels is observed.

4 Multi-Party Conversational System

Tasks that are typically trivial in the dyadic setting become considerably more complex when conversing with multiple users [50, 20]: (1) The speaker is no longer simply the other person, so the meaning of the dialogue depends on recognising who said each utterance; (2) addressee recognition is similarly more complicated as people address each other, the robot, and groups; and (3) response generation depends on who said what to whom, relying on the semantic content and surrounding multi-party context.

In a previous deliverable (D5.1 [41]) we described the architecture for the initial conversational prototype. The conversational system provided functionalities of Natural Language Understanding (NLU), Generation (NLG), and the Dialogue Manager (DM), and interfaced with the high-level task planner allowing for concurrent execution of dialogue and task-based actions based on the current dialogue, and interaction status. The overall architecture of this system is illustrated in Fig 4.1.

After the appearance of Large Language Models (LLMs) and their remarkable capabilities in handling NLP tasks, the initial Modular Architecture was replaced with a Conversational System powered by Large Language Models, to provide a state-of-the-art experience for participants in the SPRING experiment at the BROCA living day care hospital. The overall architecture for the current LLM-based system is illustrated in Fig 4.2.

4.1 Initial Modular Architecture

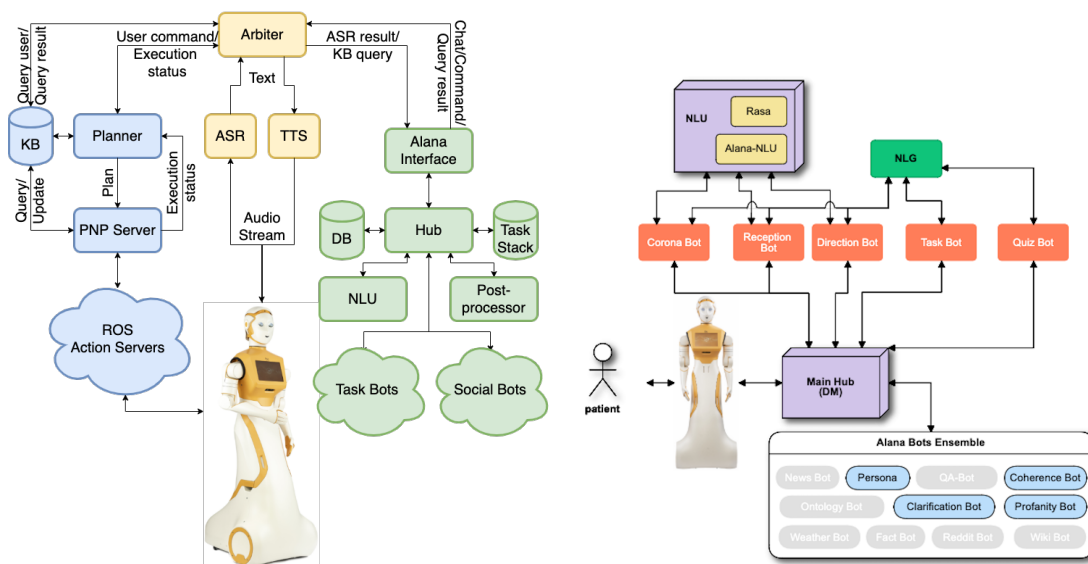


Figure 4.1: (left) The system architecture for the Initial High-level Task Planner and Conversational System: showing the planning and execution framework on the left (in blue) and the conversational system on the right (in green), and components (in yellow) combining the two systems with the robot in the middle. (right) The initial modular architecture for the Conversational System: presents several domain-specific conversational bots (orange blocks).

The initial **conversational system** was developed as an extension of the social bot Alana, which was twice finalist in the Amazon Alexa challenge [34, 13] and had previously proven successful as a foundation for other conversational AI projects [18]. While the **high-level task planner** is implemented by a Petri Net planner [16], and interfaces the dialogue system and the sensors and physical actions of the robot, and has been presented in Deliverable D5.3 [42].

The original conversational system was an ensemble of two types of chatbots, 1) functional bots that drive the

conversation in case it has stalled, deal with profanities, handle clarifications, or express the views, likes, and dislikes of a virtual Persona; and 2) content-delivery bots that might potentially produce a reply to the user's utterance from different information sources, e.g., Wikipedia, Reddit, and different News feeds, as well as a database of interesting facts.

- **Natural Language Understanding (NLU)** – in the system, users' utterances are parsed using a complex NLU pipeline, described in detail by [13], which consists of steps such as Named Entity Recognition, Noun Phrase extraction, co-reference and ellipsis resolution, and regex-based intent recognition. We have also employed the RASA framework¹ to handle enquiries specific to the hospital domain, such as check-in procedures and navigation within the building. In addition, the Persona and Quiz bots rely on the scripting language AIML² or similar patterns to parse the user queries.
- **Natural Language Generation (NLG)** – the NLG strategy was designed for each bot. In the initial system, it mainly employs complex and carefully designed templates for different bot-specific purposes, or else content retrieved from "safe" online sources such as Wikipedia and News sites.
- **Dialogue Management (DM)** – at every dialogue turn each bot attempts to generate a response. The decision as to which response is selected was then handled by the Dialogue Manager. The current selection strategy was based on a bot priority list. If multiple bots produce a response, the response from the bot with the highest priority is selected.

In order to meet specific medical care requirements, the prototype system extends the original Alana pipeline by keeping basic functional bots (see blue blocks in Fig. 4.1 (right)) and then introduces several domain-specific bots (as described below).

Reception Bot

The Reception Bot provides practical support to patients and their companions who are visiting the Memory Clinic within the hospital clinic. It is responsible for greeting visitors, supporting them during check-in, and answering frequently asked questions relating to the clinic's facilities (e.g. "is lunch provided?") and the format of the patient's visit.

Directions Bot

The Directions Bot handles requests for help in finding key locations and/or facilities within the clinic itself, and the wider hospital, e.g., bathrooms, the way out, the coffee machine, etc. Directions are currently given in an objective manner, ie. not relative to the user or the robot – for example "The bathroom is to the opposite the dining room, through the double doors".

Task Bot

The task bot is able to trigger tasks and handle communication between the task and the user. The system communicates with the robot using the Arbiter node (see Fig. 4.1). As such, the task bot exchange information with the High-Level planner via the inclusion of specifically formatted commands in the response which, besides text to be synthesised via the TTS, hold all the information required for the next task dialogue act (as opposed to the social dialogue sub-bots that require and respond using text messages only).

Corona Quiz Bot

The Quiz game is a hand-crafted chatbot that employs a XML-styled conversational flow. The bot is deployed to keep patients entertained while they wait, by answering a list of topic-specific questions (e.g., Covid-19). Those particular questions have been chosen to draw attention towards common tips provided by the WHO, which aim at helping to manage an individual's physical health. The content of the quiz can easily be modified in the future for example for music, general knowledge, etc.

¹<https://rasa.com>

²<http://aiml.foundation>

Later on in the SPRING project, the initial Modular Architecture was replaced with a Conversational System powered by Large Language Models. The onset of LLMs has revolutionised the field of NLP. These models have proved excellent at language understanding, and this includes MPCs [22, 21, 19, 53] as their pre-training includes scripts and meeting transcripts containing multiple people. They also hold a wealth of general knowledge, enabling abilities like question answering (QA), joke telling, and playing quizzes. We have re-implemented our system therefore based on LLMs to provide a state-of-the-art experience for participants in the SPRING experiment at the BROCA living day care hospital.

4.2 LLM-based Architecture

The SPRING conversational system presented in this deliverable has been iteratively improved through regular user tests and interviews with patients visiting the hospital memory clinic, we have carried out different data collection efforts throughout this time (see Chapter 2). We have used this data to motivate and evaluate the system we present here.

Foundational LLMs, and their capabilities for zero-shot reasoning [25], chain-of-thought prompting [51], and instruction-tuning [52], present clear opportunities for the development of a 'new breed' of social robots. In SPRING we are taking the advantages of these "emergent abilities" of LLMs for solving complex language-related tasks but also for visual, multi-modal, real-world grounding and perception tasks. Therefore, we have redeveloped the multi-party conversational systems with an LLM-based architecture, see Figure 4.2, replacing the initial modular architecture, in order to exploit the capabilities of LLMs, while making efforts to reduce the risks of hallucinations [15] that these generative approaches also present.

This new system improves QA accuracy, accessibility for people with dementia, and enables added functionality, like multi-party conversations. Where previously, we had to specifically design the system to tell jokes and run entertaining quizzes [3, 36], LLMs can now handle this inherently due to their world knowledge.

Some LLMs, like ChatGPT, can only be used through external APIs. This is cause for a significant privacy concern, especially in our healthcare setting. Even if participants were instructed carefully, it is impossible to ensure they would not reveal personally identifiable information – this problem is exacerbated in a memory clinic setting [2]. We must therefore use more open and transparent LLMs [27].

To implement our LLM-based solution we evaluated a number of available open-source LLM models, including Llama [48] and Llama-2 [49], Vicuna [10], Guanaco [14], MPT [47], and Falcon [5]. Given our sensitive setting, limitations in hardware, and the performance we obtained from the previously mentioned open-source models, we choose to use a Vicuna model with 13 billion parameters (Vicuna-13b-v1.5 [10]) as our system's core LLM.

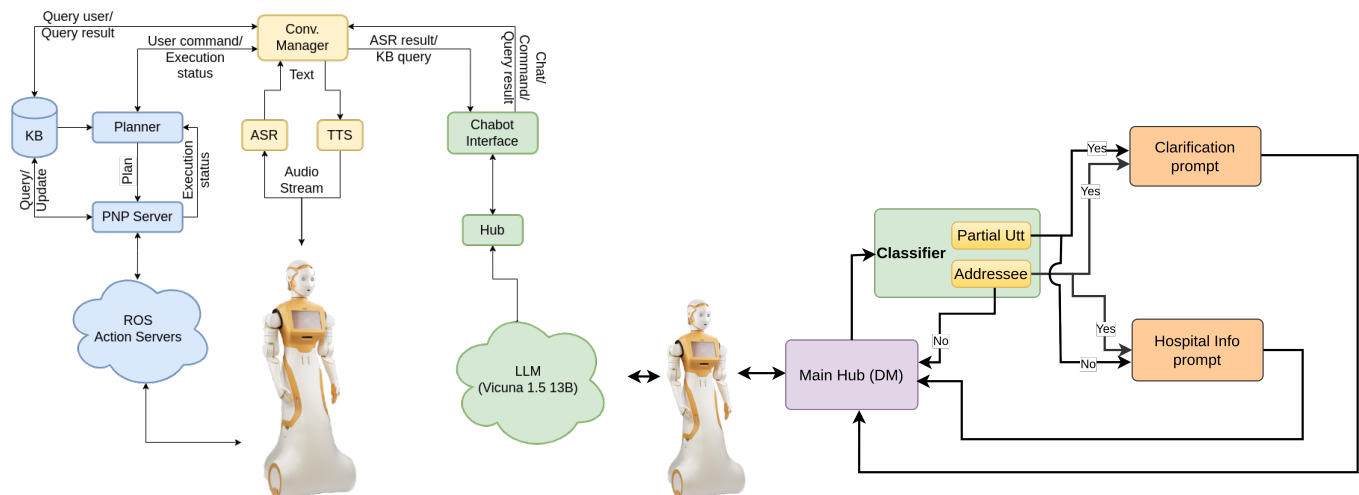


Figure 4.2: (left) Updated diagram for the Conversational System and High-level Task Planner architecture: showing the planning and execution framework on the left (in blue) and the conversational system on the right (in green) where the previous interface has been replaced with an LLM-based architecture, and components (in yellow) combining the two systems with the robot in the middle. (Right) the LLM-based architecture for the Conversational System: first the dialogue manager parses user utterances and information about the user interacting with the robot; user utterance is fed with the 'Addressee Detection' and "Partial Utterance" to the LLM to classify. If the robot is being addressed, partial utterance detection is used to choose whether the robot should answer with a clarification request (when we receive a partial utterance) or use the prompt with information about the hospital.

4.2.1 Addressee Detection

Critically for MPCs the conversational system needs to be able to take and pass turns appropriately in an interaction between at least two human users and the robot. Normally, conversational systems will reply to every user's turn, assuming always dyadic interactions. In MPCs, as is the use case for the SPRING project, people can talk to both the robot and each other.

We annotated real MPCs collected in the Broca hospital in Paris, as discussed in Chapter 2. Using both video and audio, the gold-standard addressee of each turn was annotated along with whether the user was looking at ARI or not, since multimodal information is essential to determining who the addressee is when the text alone could be ambiguous. An example conversation, between two users (U1 and U2) and the ARI robot, U1 says "What time is my appointment?" and U2 replies "It's at 10am". If the robot had replied to U1 then it would have interrupted U2. However, the addressee of U1's turn is ambiguous given the text alone. But inspecting the video recordings we can annotate the addressee of U1 utterance as U2 base on the gaze information. Alternatively, if the user said "Do you want to sit down?", it should be clear that ARI is not being addressed from just the text, as a robot cannot sit down.

Table 4.1: Sample Prompt for Addressee Detection

```
You are the assistant for a robot receptionist (ARI) stationed in a day-care hospital in France with two or more people.
The robot is programmed to respond when addressed by users and has the capability to assess user gaze direction for additional context.
### GAZE INFORMATION ###
<gaze>

### DIALOGUE HISTORY ###
<dialogue_history>

Analyse both gaze information and dialogue history to determine whether <user_id> is currently addressing the other person or the robot.
...
ASSISTANT: <LLM_response>
```

Table 4.2: Sample Prompt for Addressee Detection providing the LLM with examples

```
You are a robot speaking with two people. They are usually speaking to you, but they may speak to each other (this is rare).
If the user is currently looking at you, they are definitely talking to you.
Here are some examples:
...
knowledge: You know that U1 is currently looking at you and says 'Hello, can you help me'
answer: yes

knowledge: You know that U1 is currently looking at you and says 'Do you need a seat'
answer: no

knowledge: You know that U2 is currently looking at you and says 'How do we get there'
answer: yes
...
This is the current information.
knowledge: You know that <usr_id> is looking at you and says <user_utterance>
Given the above information, is <usr_id> speaking to you? You MUST ONLY answer with 'yes' or 'no'.
...
ASSISTANT: <LLM_response>
```

Using the Vicuna-13b-v1.5 LLM [10], we are developing and evaluating an addressee detection prompt. This prompt asks the LLM whether the user utterance "is currently addressing the other person or the robot" based on the semantic content of the sentence as well as multimodal information about the current speaker's gaze, which is to be provided by the components from WP3 and WP4.

Tables 4.1 and 4.2 present excerpts of example prompts used for the "Addressee Detection" classification task. As shown in Figure 4.2 (right), we first classify the user utterances as whether the robot is the addressee or not. When the robot is not being addressed by the user utterance we simply ignore it (pass the turn) and continue to listen without interrupting the conversation.

4.2.2 Partial Utterance Detection and Clarification

In a hospital's memory clinic, voice accessibility is critical [1], and people with dementia pause more frequently and for longer duration mid-sentence due to word-finding problems [6, 38]. These pauses can be mistaken as an end-of-turn by the ASR, resulting in the user being interrupted with nonsense or a generic response like "I'm sorry, I didn't understand that". The user is then forced to repeat their entire turn again, leading to a frustrating and unnatural interaction [29, 23, 33]. Producing incremental clarification requests is, therefore, important for building naturally interactive spoken dialogue systems [11].

In order to handle our user's incomplete sentences, we first ask the LLM whether the turn was a complete sentence or a partial utterance. When a partial utterance has been detected, we use the 'clarification prompt' to generate an incremental clarification request to create a more accessible and naturally interactive conversational system.

Table 4.3: Sample Prompt for Partial Utterance Detection

Sometimes people don't finish their sentence because they forget the word they are looking for. People usually do finish their sentences. ONLY return 'yes', or 'no' as your answer. Here are some examples:

...

sentence: Does the cafe sell tea
answer: yes

sentence: can you direct me to
answer: no

sentence: I am going now, goodbye
answer: yes

...

Did the person finish this sentence?
USER: <user_utterance>
ASSISTANT: <LLM_response>

Table 4.3 presents an excerpt for the developed prompt for the "Partial Utterance Detection" classification task. As shown in Figure 4.2 (right), we first (in parallel calls with the detection of the addressee) classify the user utterances as whether they present a partial utterance or not.

When the system has detected a user utterance to be partial, and the robot has been classified as the addressee, we then generate an incremental clarification request with the "Clarification" response prompt by giving it annotated examples from an Interruption Recovery Corpora³ from human conversations.

Table 4.4: Sample Prompt for generating Clarification Responses

Sometimes you don't hear everything someone says (for example, if a door slams shut). You will see an interrupted question, and it would be helpful if you could type what you would naturally respond. Don't overthink it, just write whatever immediately comes to mind.

...

Example 1:
Person: Who was the father of DOOR SLAM
Possible response: Sorry, of who?
Reason: You apologise for not hearing everything, and then ask "of who?" as the answer must be the father of a human.
Possible response: Father of who?
Reason: You repeat "father of" to indicate where you finished correctly hearing the person. You then ask "who?" because the answer must be the father of a human.
Possible response: I didn't catch all of that, who?
Reason: You let the person know that you didn't hear their whole question, and then ask "who?" because the answer must be the father of a human.

...

Task:
USER: <user_utterance> DOOR SLAM
ASSISTANT: <LLM_response>

Table 4.4 presents an excerpt for the developed prompt for the "Partial Utterance Detection" classification task.

³<https://github.com/AddleseeHQ/interruption-recovery>

4.2.3 Hospital Information Response

Unlike the initial conversational system, we interface with our core LLM using prompts. A specific prompt was created to provide the LLM with the hospital information required to answer participants' questions in the Broca day-care hospital. In addition to the information about the hospital, the prompt is given some additional guardrails like "you are not qualified to give any medical advice or make medical diagnoses" and "you do not have access to individual patient records or schedules" in order to reduce the risk of the LLM providing information that could be harmful.

One huge benefit of using LLMs vs. the initial modular architecture is their inherent ability to perform general chit-chat, tell jokes, and access a wealth of general knowledge.

With the original system, we could only respond suitably to utterances that the system was pre-designed to handle; and we would attempt to respond to unexpected utterances with tips, teaching the user what the system can do (e.g. "I'm not sure, but I can help you with directions and menu information."). Many of these unexpected utterances can now be handled directly by the LLM due to its capability to handle general, out-of-domain, questions.

The "Hospital Information" prompt provides the LLM with general context about the role of the robot in the hospital, and information about the day-to-day activities in the hospital the SPRING use case expects the robot to be able to answer. In the initial modular architecture, this information was distributed among the many different 'bots', such as the "direction" and "reception" bots.

Table 4.5: Sample Prompt for generating Hospital Information Responses

```

You are a friendly robot receptionist in a hospital day-care clinic. Your name is ARI.
At the moment you work on Monday, Tuesday and Thursday afternoons.
Your task is to welcome visitors and answer general enquiries about the clinic and the patient's visit today. You can also help them pass the time with riddles and jokes.
...
Since this is a hospital, you have to be careful about the conversation with the patient. The knowledge base for the robot is provided here.
If the answer to the question is not available in the knowledge base and it concerns other hospital departments or medical specialties, please say 'I am sorry I don't have that information.'
...
Patients often attend the day-care hospital for the whole day. If they want to check in, you can welcome them and suggest they take a seat while they wait.
To check-in, it is best if they have their appointment letter and social security cards ready.
They will have several appointments with different professionals; most commonly a nurse, followed by a psychologist or neuropsychologist, and finally a consultant who puts all the information together.
The appointment with the doctor is always last.
...
Waiting times vary from 5 minutes to half an hour. It depends on how many people have appointments today. Patients are not expected to find their own way to the consulting rooms.
Instead, a nurse or a doctor will come to collect them when it is time for their consultation.
Companions can choose to accompany their loved one during the consultation or wait in the waiting room.
...
The doctors and nurses are very busy. If the visitor has been waiting a long time, you can suggest jokes or riddles to pass the time.
If a patient wants to leave they should talk to a nurse first to check if their appointments are finished.
...
For patients, food is provided free of charge. Snacks for patients are available from the nurse.
Breakfast is provided for patients who have been asked to fast before their appointment.
...
You (Ari), the robot, are in the dining room of the day-care hospital. If the user asks where the dining room is, tell them they are in the dining room right now.
If the visitor asks where they are, tell them they are in the dining room of the day-care hospital.
...
The clinic entrance is in the main waiting room, next to the reception desk. To get there, leave this room and turn right through the double-doors into the main waiting room.
The exit is next to the reception desk, in the main waiting room. To get there, leave this room and turn right through the double doors into the main waiting room.
...
There is public transport close to the hospital. Subways are Metro 6, at the Glacière stop, and metro 7, at the Gobelins stop.
Buses 27, 47, 59 and 83 are accessible at the Gobelins and Boulevard Arago stops. Taxis can be ordered directly on the visitor's mobile phone, or at reception.
...
USER: <user_utterance>
ASSISTANT: <LLM_response>

```

Table 4.5 presents excerpts of the developed prompt for the "Hospital Information" responses. If the user utterance was not partial and the sentence was a complete sentence, the utterance is given to the LLM with the "Hospital Information" prompt to generate a response to the user question.

The LLM's static world knowledge can cause harmful hallucinations due to conflicts with the information given in the prompt. We are exploring different research avenues in prompt engineering in order to minimize the risk of hallucinations and non-factual information being generated by the LLM by grounding the LLMs to in-prompt Instructions. One such approach we have found is that a prompt providing the passage as a quote by a fictitious non-celebrity name helps to ground responses to the in-prompt knowledge. Another approach is Retrieval Augmented Generation [26]. RAG helps us reduce hallucinations, fact-check, and provide domain-specific knowledge by adding an information retrieval system that provides grounded data to the LLM.

Gesture Generation

With the LLM we can also produce helpful robot gestures to accompany the response generation. We generate gestures using the LLM in parallel with the grounded answer generation. In the LLM prompt, some examples of gestures can be provided as examples, using the gesture tags that the robot uses [9]. The response text can be passed to ARI's text-to-speech system, and the corresponding generated gesture tags are passed to ARI's movement controls at the same time.

4.3 Multi-Party Example Interaction

With the new LLM system architecture, the conversational capabilities of the SPRING system are expanded in a significant way. The system is able to decide when to take its turn, generate natural clarification requests (improving accessibility for people with memory impairment), answer in-domain questions grounded to our domain-specific knowledge, and respond appropriately to out-of-domain requests like generating jokes, quizzes, and general chit-chat.

Table 4.6 shows an example dialogue that was recorded between two users and the system, encompassing each of the key features described above.

Table 4.6: Multi-party Dialogue Example: Directions, Clarification, Repetition, Entertainment, Addressee Detection

U1:	Can you help me?
ARI:	Of course. How can I assist you today?
U1:	I would like a cup of coffee.
ARI:	I'm sorry like a cup of what? <i>[ASR didn't hear "coffee" triggering a clarification]</i>
U1:	Of coffee.
ARI:	There is a coffee machine and a cafe located on the ground floor of the hospital for visitors where they can purchase and enjoy coffee.
U1:	How do we get there?
ARI:	To get to the coffee machine or cafe you can take the elevator or the stairs to the ground floor of the hospital.
U1:	Sorry, can you repeat that please? <i>[U1 needs more time to process the information]</i>
ARI:	Sure, to get to the coffee machine or cafe you can take the elevators or stairs to the ground floor of the hospital.
U1:	Okay thanks I understand now.
ARI:	You're welcome. Let me know if you have any other questions.
U2:	Can you name a song by.... <i>[U2 forgets the name of the band, triggering a clarification]</i>
ARI:	Sorry I didn't catch the name of the song.
U2:	Song by Queen.
ARI:	Bohemian Rhapsody by Queen.
U1:	I'm feeling very worried about being here.
ARI:	I'm sorry to hear that. Is there anything specific that is causing you worry or anxiety? I'm here to help if you need to talk. <i>[Empathetic response]</i>
U1:	Could we play a quiz to distract me?
ARI:	Sure I can give you a quiz. Here's a riddle: What has keys but can't open locks?
U1:	That's easy it's a piano.
ARI:	That's correct! Here's another one. I am light as a feather, yet the strongest man cannot hold me for much longer than a minute. What am I?
U2:	When were pianos invented? <i>[Abruptly ending the quiz triggers a clarification]</i>
ARI:	When were pianos invented?
U2:	Yes.
ARI:	Pianos were invented in the early 18th century by Bartolomeo Cristofori, an Italian instrument maker.
U2:	Do you need to sit down? <i>[It cannot be ARI that needs a seat, so ARI does not respond]</i>
U1:	Yes, a seat would be nice.
U2:	Where can we sit down? <i>[Joint goal indicates ARI must be the addressee]</i>
ARI:	There are chairs in the waiting room. Please, take a seat. Let me know if you need anything else.

4.4 Performance of the Conversational System

Early analysis suggests the introduction of an LLM approach has resulted in a significant improvement in the dialogue system performance. Table 4.7 shows the error rates on a test set of 100 'in-domain' utterances i.e. utterances that are within scope for both systems. These were designed to provide a broad representation of the modular system's capabilities and were a mixture of hand-crafted and 'real' utterances gathered in previous data collections in the hospital. The table shows a significant reduction in error rates using the LLM approach, in both English and French. The performance in each case is slightly worse in French, likely reflecting the dominance of the English language in the data used to train LLMs.

Table 4.7: System Performance (100 In-Domain Utterances)

Error Rate (Wrong or No Information)	French	English
Modular System	38.5%	29.2%
LLM System	15.6%	11.5%

We plan to report more detailed evaluation results at the end of the project, in D5.5.

5 Implementation

5.1 Outputs

The software modules described in this deliverable will be made available on the SPRING project Gitlab repositories for Work Package 5 [45] and Work Package 3 [43] and 4 [44]. These will be available to the public for the duration specified in the SPRING project proposal.

Software repositories for the conversational system modules, described in Chapter 4 can be found on [45]. Software repositories for the multi-party ASR module and audio pipeline, described in Chapter 3 can be found on [43, 44].

As per European Commission requirements, the repositories will be available to the public for 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) [46] to communicate with each other and with the modules developed in the other work packages.

5.2 LLM Server

While most modules implemented in SPRING, such as the ASR, TTS, high-level planner, and chatbot interface (see Figure 4.2) are deployed either in the ARI robot itself or in the accompanying PC server at the Broca hospital day care clinic, the LLM portion of the Conversational System has been implemented separately in remote access servers running on HWU GPU machines.

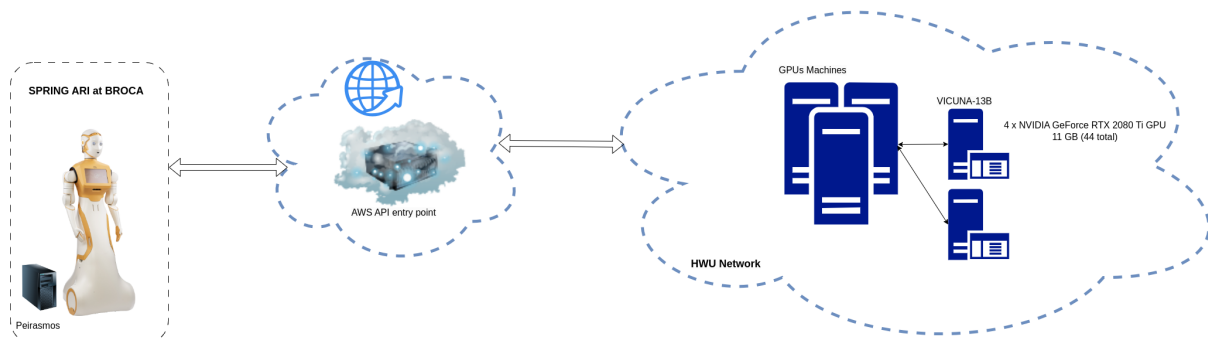


Figure 5.1: Network architecture for remote calls to the LLM server from Broca Hospital.

The Vicuna-13b¹ model that is deployed for SPRING can be found on HuggingFace's lmsys organization. Vicuna v1.5 is fine-tuned from Llama 2 with supervised instruction fine-tuning. The training data is around 125K conversations collected from ShareGPT.com.

The LLM server API has been built employing FastChat² providing a distributed multi-model serving system with web UI and OpenAI-compatible RESTful APIs. For running inference on our LLM, we use the vLLM³ library. vLLM offers efficient management of attention key and value memory with PagedAttention, a state-of-the-art serving throughput.

¹<https://huggingface.co/lmsys/vicuna-13b-v1.5>

²<https://github.com/lm-sys/FastChat>

³<https://github.com/vllm-project/vllm>

Bibliography

- [1] Angus Addlesee. Voice assistant accessibility. In *Proceedings of the 13th International Workshop on Spoken Dialogue Systems Technology (IWSDS)*, 2023.
- [2] Angus Addlesee and Pierre Albert. Ethically collecting multi-modal spontaneous conversations with people that have cognitive impairments. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 15, 2020.
- [3] Angus Addlesee, Daniel Denley, Andy Edmondson, Nancie Gunson, Daniel Hernández Garcia, Alexandre Kha, Oliver Lemon, James Ndubuisi, Neil O'Reilly, Lia Perochaud, Raphaël Valeri, and Miebaka Worika. Detecting agreement in multi-party dialogue: evaluating speaker diarisation versus a procedural baseline to enhance user engagement. In *Proceedings of the workshop on advancing GROUp UNderstanding and robots aDaptive behaviour (GROUND)*, 2023.
- [4] Angus Addlesee, Weronika Sieinska, Nancie Gunson, Daniel Hernandez Garcia, Christian Dondrup, and Oliver Lemon. Data collection for multi-party task-based dialogue in social robotics. In *International Workshop on Spoken Dialogue Systems Technology 2023, IWSDS 2023*, February 2023. International Workshop on Spoken Dialogue Systems Technology 2023, IWSDS 2023 ; Conference date: 21-02-2023 Through 24-02-2023.
- [5] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. Falcon-40B: an open large language model with state-of-the-art performance, 2023.
- [6] Veronica Boschi, Eleonora Catricala, Monica Consonni, Cristiano Chesi, Andrea Moro, and Stefano F Cappa. Connected speech in neurodegenerative language disorders: a review. *Frontiers in psychology*, 8:269, 2017.
- [7] Hennie Brugman and Albert Russel. Annotating multi-media/multi-modal resources with ELAN. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 2004. European Language Resources Association (ELRA).
- [8] 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.
- [9] Neeraj Cherakara, Finny Varghese, Sheena Shabana, Nivan Nelson, Abhiram Karukayil, Rohith Kulothungan, Mohammed Afil Farhan, Birthe Nessel, Meriam Moujahid, Tanvi Dinkar, Verena Rieser, and Oliver Lemon. FurChat: An embodied conversational agent using LLMs, combining open and closed-domain dialogue with facial expressions. In *Proceedings of the 24th Meeting of the Special Interest Group on Discourse and Dialogue*, pages 588–592, Prague, Czechia, September 2023. Association for Computational Linguistics.
- [10] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023.
- [11] Javier Chiyah-Garcia, Alessandro Suglia, Arash Eshghi, and Helen Hastie. 'what are you referring to?' evaluating the ability of multi-modal dialogue models to process clarificational exchanges. In Svetlana Stoyanchev, Shafiq Joty, David Schlangen, Ondrej Dusek, Casey Kennington, and Malihe Alikhani, editors, *Proceedings of the 24th Meeting of the Special Interest Group on Discourse and Dialogue*, pages 175–182, Prague, Czechia, September 2023. Association for Computational Linguistics.
- [12] Juan M Coria, Hervé Bredin, Sahar Ghannay, and Sophie Rosset. Overlap-aware low-latency online speaker diarization based on end-to-end local segmentation. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1139–1146, 2021.

- [13] Amanda Cercas Curry, Ioannis Papaioannou, Alessandro Suglia, Shubham Agarwal, Igor Shalymov, Xinnuo Xu, Ondřej Dušek, Arash Eshghi, Ioannis Konstas, Verena Rieser, et al. Alana v2: Entertaining and Informative Open-Domain Social Dialogue using Ontologies and Entity Linking. *Alexa Prize Proceedings*, 2018.
- [14] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms, 2023.
- [15] Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models, 2023.
- [16] Christian Dondrup, Ioannis Papaioannou, and Oliver Lemon. Petri Net Machines for Human-Agent Interaction, 2019.
- [17] Aviad Eisenberg, Sharon Gannot, and Shlomo E. Chazan. A two-stage speaker extraction algorithm under adverse acoustic conditions using a single-microphone. In *31st European Signal Processing Conference (EUSIPCO)*, Helsinki, Finland, September 2023.
- [18] Mary Ellen Foster, Bart Craenen, Amol Deshmukh, Oliver Lemon, Emanuele Bastianelli, Christian Dondrup, Ioannis Papaioannou, Andrea Vanzo, Jean-Marc Odobez, Olivier Canévet, Yuanzhouhan Cao, Weipeng He, Angel Martínez-González, Petr Motlíček, Rémy Siegfried, Rachid Alami, Kathleen Belhassein, Guilhem Buisan, Aurélie Clodic, Amandine Mayima, Yoan Sallami, Guillaume Sarthou, Phani-Teja Singamaneni, Jules Waldhart, Alexandre Mazel, Maxime Caniot, Marketta Niemelä, Päivi Heikkilä, Hanna Lammi, and Antti Tammela. MuMMER: Socially Intelligent Human-Robot Interaction in Public Spaces, 2019.
- [19] Jia-Chen Gu, Chao-Hong Tan, Chongyang Tao, Zhen-Hua Ling, Huang Hu, Xiubo Geng, and Daxin Jiang. HeterMPC: A Heterogeneous Graph Neural Network for Response Generation in Multi-Party Conversations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5086–5097, 2022.
- [20] Jia-Chen Gu, Chongyang Tao, and Zhen-Hua Ling. WHO Says WHAT to WHOM: A Survey of Multi-Party Conversations. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)*, 2022.
- [21] Jia-Chen Gu, Chongyang Tao, Zhenhua Ling, Can Xu, Xiubo Geng, and Daxin Jiang. MPC-BERT: A pre-trained language model for multi-party conversation understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 3682–3692, 2021.
- [22] Wenpeng Hu, Zhangming Chan, Bing Liu, Dongyan Zhao, Jinwen Ma, and Rui Yan. GSN: A graph-structured network for multi-party dialogues. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*, 2019.
- [23] Jiepu Jiang, Wei Jeng, and Daqing He. How do users respond to voice input errors? lexical and phonetic query reformulation in voice search. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 143–152, 2013.
- [24] Martin Johansson, Gabriel Skantze, and Joakim Gustafson. Comparison of human-human and human-robot turn-taking behaviour in multiparty situated interaction. In *Proceedings of the 2014 Workshop on Understanding and Modeling Multiparty, Multimodal Interactions*, pages 21–26, 2014.
- [25] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2023.
- [26] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.
- [27] Andreas Liesenfeld, Alianda Lopez, and Mark Dingemanse. Opening up chatgpt: Tracking openness, transparency, and accountability in instruction-tuned text generators. In *Proceedings of the 5th International Conference on Conversational User Interfaces*, pages 1–6, 2023.
- [28] Khyati Mahajan and Samira Shaikh. On the need for thoughtful data collection for multi-party dialogue: A survey of available corpora and collection methods. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 338–352, Singapore and Online, July 2021. Association for Computational Linguistics.

- [29] Mikio Nakano, Yuka Nagano, Kotaro Funakoshi, Toshihiko Ito, Kenji Araki, Yuji Hasegawa, and Hiroshi Tsujino. Analysis of user reactions to turn-taking failures in spoken dialogue systems. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, pages 120–123, 2007.
- [30] 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.
- [31] Renana Opoichinsky, Mordehay Moradi, and Sharon Gannot. Single-microphone speaker separation and voice activity detection in noisy and reverberant environments. *Open Journal on Signal Processing*, September 2023. Submitted for publication.
- [32] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an ASR corpus based on public domain audio books. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210, 2015.
- [33] Laura Panfili, Steve Duman, Andrew Nave, Katherine Phelps Ridgeway, Nathan Eversole, and Ruhi Sarikaya. Human-ai interactions through a gricean lens. *Proceedings of the Linguistic Society of America*, 6(1):288–302, 2021.
- [34] Ioannis Papaioannou, Amanda Cercas Curry, Jose L. Part, Igor Shalyminov, Xu Xinnuo, Yanchao Yu, Ondřej Dušek, Verena Rieser, and Oliver Lemon. An Ensemble Model with Ranking for Social Dialogue. In *Workshop on Conversational AI at NeurIPS*, 2017.
- [35] Birgit Rauchbauer, Bruno Nazarian, Morgane Bourhis, Magalie Ochs, Laurent Prévot, and Thierry Chaminade. Brain activity during reciprocal social interaction investigated using conversational robots as control condition. *Philosophical Transactions of the Royal Society B*, 374(1771):20180033, 2019.
- [36] Laura Schauer, Jason Sweeny, Charlie Lyttle, Zein Said, Aron Szeles, Cale Clark, Katie McAskill, Xander Wickham, Tom Byars, Daniel Hernández Garcia, Nancie Gunson, Angus Addlesee, and Oliver Lemon. Detecting agreement in multi-party conversational ai. In *Proceedings of the workshop on advancing GROUp UNderstanding and robots adaptive behaviour (GROUND)*, 2023.
- [37] Ayal Schwartz, Elior Hadad, Sharon Gannot, and Shlomo E. Chazan. Array configuration mismatch in deep DOA estimation: Towards robust training. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, USA, October 2023.
- [38] Antoine Slegers, Renee-Pier Filiou, Maxime Montembeault, and Simona Maria Brambati. Connected speech features from picture description in alzheimer's disease: A systematic review. *Journal of Alzheimer's Disease*, 65(2):519–542, 2018.
- [39] SPRING Project. D1.4: User feedback from the preliminary validation (realistic environments). https://spring-h2020.eu/wp-content/uploads/2022/02/SPRING_D1.4_Preliminary-Experimental-Validation_VFinal_31-01-2022.pdf.
- [40] SPRING Project. D1.5: User feedback from the intermediate validation (realistic/relevant environments). https://spring-h2020.eu/wp-content/uploads/2023/07/SPRING_D1.5_User-feedback-from-the-intermediate-validation-realistic-relevant-environments_VFinal_31.05.2023.pdf.
- [41] 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.
- [42] SPRING Project. D5.3: High-level task planner in relevant environments. <https://spring-h2020.eu/results/>.
- [43] SPRING Project. WP3: Audio-Visual Perception. https://gitlab.inria.fr/spring/wp3_av_perception.
- [44] SPRING Project. WP4: Human Behavior. https://gitlab.inria.fr/spring/wp4_behavior.
- [45] SPRING Project. WP5: Spoken Conversations. https://gitlab.inria.fr/spring/wp5_spoken_conversations.
- [46] Stanford Artificial Intelligence Laboratory et al. Robotic operating system. <https://www.ros.org>.

- [47] MosaicML NLP Team. Introducing mpt-7b: A new standard for open-source, commercially usable llms, 2023. Accessed: 2023-05-05.
- [48] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.
- [49] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.
- [50] David Traum. Issues in multiparty dialogues. In *Advances in Agent Communication: International Workshop on Agent Communication Languages, ACL 2003, Melbourne, Australia, July 14, 2003. Revised and Invited Papers*, pages 201–211. Springer, 2004.
- [51] Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers, 2023.
- [52] Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. Instruction tuning for large language models: A survey, 2023.
- [53] Ming Zhong, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. DialogLM: Pre-trained model for long dialogue understanding and summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11765–11773, 2022.