HOTSPOT: An Ad Hoc Teamwork Platform for Mixed Human-Robot Teams

João G. Ribeiro [1], Luis Müller Henriques [1], Sérgio Colcher [1], Julio Cesar Duarte [2], Francisco S. Melo [1], Ruy Luiz Milidiú [1], and Alberto Sardinha [1]

[1]Affiliation not available
[2]Instituto Militar de Engenharia

October 30, 2023

Abstract

Ad hoc teamwork is a research topic in multi-agent systems whereby an agent (the "ad hoc agent") must successfully collaborate with a set of unknown agents (the "teammates") without any prior coordination or communication protocol. However, research in ad hoc teamwork is predominantly focused on agent-only teams, but not in agent-human teams, which we believe is an exciting research avenue and has enormous application potential in human-robot teams. This paper will tap into this potential by proposing HOTSPOT, the first framework for ad hoc teamwork in human-robot teams. Our framework comprises two main modules, addressing the two key challenges in the interaction between a robot acting as the ad hoc agent and human teammates. First, a *decision-theoretic module* that is responsible for all task-related decision making (task identification, teammate identification, and planning). Second, a *communication module* that uses natural language processing in order to parse all communication between the robot and the human. To evaluate our framework, we use a task where a mobile robot and a human cooperatively collect objects in an open space, illustrating the main features of our framework in a real-world task.

HOTSPOT: An Ad Hoc Teamwork Platform for Mixed Human-Robot Teams

João G. Ribeiro, Luis Müller Henriques, Sérgio Colcher, Julio Cesar Duarte, Francisco S. Melo, Ruy Luiz Milidiú, and Alberto Sardinha

***Abstract*—Ad hoc teamwork is a research topic in multi-agent systems whereby an agent (the "ad hoc agent") must successfully collaborate with a set of unknown agents (the "teammates") without any prior coordination or communication protocol. However, research in ad hoc teamwork is predominantly focused on agent-only teams, but not in agent-human teams, which we believe is an exciting research avenue and has enormous application potential in human-robot teams. This paper will tap into this potential by proposing HOTSPOT, the first framework for ad hoc teamwork in human-robot teams. Our framework comprises two main modules, addressing the two key challenges in the interaction between a robot acting as the ad hoc agent and human teammates. First, a *decision-theoretic module* that is responsible for all task-related decision making (task identification, teammate identification, and planning). Second, a *communication module* that uses natural language processing in order to parse all communication between the robot and the human. To evaluate our framework, we use a task where a mobile robot and a human cooperatively collect objects in an open space, illustrating the main features of our framework in a real-world task.**

***Index Terms*—Ad Hoc Teamwork, Multi-Agent Systems, Human-Robot Interaction, Natural Language Processing.**

I. Introduction

RECENT decades have witnessed a significant shift in the use of robots. While robotic platforms still find extensive use in industry, advances in hardware and software have enabled the development of various robotic platforms for everyday use. For instance, robots are being used in healthcare [1], assisted living [2], entertainment [3], and even for mundane tasks such as cleaning [4]. Moreover, as the use of robots broadens beyond industrial applications, the ability of these platforms to naturally interact with users that have no technical expertise becomes a mandatory requirement. Thus, it is not surprising that the area of human-robot interaction has seen impressive growth in the last decades.

In this paper, we are particularly interested in *collaborative* human-robot interaction. This topic is not novel, and a significant body of literature has investigated human-robot collaboration from many different perspectives [5]. However, very few works focus on the problem of *ad hoc teamwork* involving a human and a robot.

Ad hoc teamwork was proposed originally in the multi-agent systems community [6] and addresses the problem of an agent (henceforth called the "ad hoc agent") that must successfully cooperate with a group of unknown "teammates"—i.e., other agents about which the ad hoc agent has little or no information. This group of agents must now act as a *team*, even if they have no prior cooperation or coordination mechanisms. The role of the ad hoc agent is to understand or infer *what* is the task that the other agents are performing, *who* among the other agents is doing what, towards the completion of the task, and then decide *how* to contribute. These three challenges were identified in Melo and Sardinha [7] as fundamental sub-tasks of the ad hoc teamwork problem, dubbed *task identification*, *teammate identification* and *planning*.

So far, research in ad hoc teamwork has focused primarily on agent-agent interaction scenarios [8], and rests on strong assumptions regarding what the ad hoc agent is able to perceive regarding the environment, the teammates, and/or the task to be addressed. Dealing with teams of humans and robots brings forth several critical challenges that current research on ad hoc teamwork has not considered yet. For example:

- Robots, as embodied agents, have to deal with perceptual and actuation challenges that virtual agents seldom consider. In particular, the robot's perception of its state is often imperfect, and its actuation is prone to failures;
- The teammate—being a human—does not behave according to a well-defined model (for example, it is not necessarily optimal or rational);
- Decision-making must be conducted at run-time.

These challenges are common in human-robot interaction scenarios but rarely considered in the ad hoc teamwork literature (if at all). Additionally, humans can communicate through natural language, and such communication channels can be rich and informative if the robot can take advantage of them. However, dealing with natural language is another challenge

Manuscript received November 2021; revised 2021.

This work was partially supported by national funds through FCT, Fundação para a Ciência e a Tecnologia, under project UIDB/50021/2020 (INESC-ID multi-annual funding) and the HOTSPOT project, with reference PTDC/CCI-COM/7203/2020. This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-19-1-0020, and by TAILOR, a project funded by EU Horizon 2020 research and innovation programme under GA No 952215. João G. Ribeiro acknowledges the PhD grant 2020.05151.BD from FCT. (Corresponding author: Julio Cesar Duarte)

S. Colcher, J. Duarte, R. Milidú and L. Henriques are with Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, R. Marquês de São Vicente, 225 - Gávea, Rio de Janeiro - RJ, 22541-041, Brazil, e-mail: {colcher, jduarte, milidiu, lhenriques}@inf.puc-rio.br.

F. Melo. A. Sardinha and J. Ribeiro are with INESC-ID, Instituto Superior Técnico, Universidade de Lisboa, Av. Prof. Dr. Cavaco Silva, Taguspark 2744-016 Porto Salvo, Portugal, e-mail: fmelo@inesc-id.pt, {jose.alberto.sardinha, joao.g.ribeiro}@tecnico.ulisboa.pt.

J. Duarte is also with Seção de Ensino de Engenharia da Computação, Instituto Militar de Engenharia, Praça Gen. Tibúrcio - 80, Urca, Rio de Janeiro - RJ, 22290-270, Brazil, e-mail: duarte@ime.eb.br.

for robot developers, although the potential applicability of ad hoc teamwork in everyday tasks is enormous and mostly untapped.

This paper's main contribution is a framework for ad hoc teamwork between a human and a robot. Our framework, which we dub HOTSPOT[1], instantiates the problem of ad hoc teamwork in human-robot teams as follows. A human and a robot co-exist in a shared environment and must perform a collaborative task that requires them to coordinate their actions. The human knows the task, but the robot (the ad hoc agent in our setting) does not. From observations of the human, the robot must infer the task (among a set of possible tasks), understand how the human user is performing it, and adapt its decision process towards completing the task.

In our scenario, we consider tasks in an open space (i.e., a lab), where the human and the robot have to move around. This requirement poses additional challenges to the robot due to the semi-unstructured interaction, where the robot will not know the location of the human user most of the time. To address this ad hoc teamwork problem, we propose a *decision-theoretic approach* that extends that of Ribeiro et al. [9] to account for the perceptual limitations of the ad hoc agent (the robot). At the same time, to take advantage of the fact that the human can communicate using natural language, we endow the robot with the ability to *communicate* with the human user. In particular, the robot can communicate through spoken utterances, querying the user about the task's current state, interpreting the human's response using natural language processing (NLP), and considering the inherent uncertainty in that interpretation process.

We evaluate the HOTSPOT framework in a real-world scenario involving the interaction between a human and a robot in an open space, evaluating the performance of each module individually and of the whole framework.

To summarize, the contributions of this work are three-fold:

- We contribute a *decision-theoretic model for ad hoc teamwork with limited perception*. We describe the decision problem faced by the ad hoc agent (the robot) as a *partially observable Markov decision problem* and use a standard heuristic solution to compute an adequate policy for the robot efficiently. Our results show that our approach can infer and complete the task in a near-optimal number of steps while still using partial and imperfect information.
- We contribute a *natural language processing model* that allows the robot to understand the utterances issued by the human, use that information to locate itself in the environment, and express itself in an easy way that the human can understand. Our results show that the full NLP models achieved an accuracy of about 80% in every task performed.
- We contribute an empirical validation of our approach in a real-world ad hoc teamwork scenario involving a human and a mobile robot.

The paper is organized as follows. Section II provides an overview of related work in both ad hoc teamwork and NLP, framing the contributions of this paper in the landscape of existing research. Section III provides an in-depth description of the HOTSPOT framework, describing its two main modules and how they interact. Section IV describes the procedure used to validate our framework. Finally, section V discusses the results of our evaluation, and Section VI concludes.

[1]HOTSPOT stands for "Human-robOt TeamS without PrecoOrdinaTion".

II. Related Work

This section frames our contribution in the context of existing research both on *ad hoc teamwork* and *natural language processing*, since these are the two areas of research most relevant for our present work.

Regarding ad hoc teamwork, the problem was originally proposed in the pioneer work of Stone et al. [6], and has spanned a significant volume of research [7, 9–11]. Following Melo and Sardinha [7], we can break down the ad hoc teamwork problem into three main sub-problems: *task identification*, *teammate identification*, and *planning*.

Early research into ad hoc teamwork focused on the *planning* step. For example, Stone and Kraus [12] proposed one of the first planning algorithms for ad hoc teamwork, by formulating the problem as a cooperative k-armed bandit with known teammates. Similarly, Stone et al. [13] and Agmon and Stone [14] look at ad hoc teamwork as a problem of "leading" known teammates to perform actions that yield optimal joint performance.

Barrett et al. [8] introduce the PLASTIC framework to address ad hoc teamwork when facing both unknown task and teammates, using a reinforcement learning approach. To this day, the PLASTIC algorithms remain among the state-of-the-art in ad hoc teamwork, addressing both *task and teammate identification*.

Melo and Sardinha [7] address the three sub-problems of ad hoc teamwork by proposing two distinct approaches. Specifically, one approach is based on sequential prediction [15] and the other one on decision-theoretic planning [16]. The two approaches consider one-shot problems and, as such, are not suited for sequential problems. Along the same lines, Ribeiro et al. [9] extend the previous work to address sequential tasks under uncertainty.

Most previous works, however, consider only agent-agent interaction in that they disregard critical difficulties found when an embodied agent (such as a robot) must interact with human teammates. A human-robot interaction setting must consider the limitations of using a robotic platform (such as unreliable and limited perception, unreliable actuation) and the interaction with a human user.

Nevertheless, some recent works take important steps towards bringing ad hoc teamwork closer to human-robot interaction scenarios. For example, in terms of ad hoc teamwork involving robots, Genter et al. [17] investigated the use of ad hoc algorithms in the RoboCup World Championship, in the context of the Drop-in Player Competition [18].

Fern et al. [19] address the problem of *assistance* which, although not formulated as an ad hoc teamwork problem, shares many of its challenges. Specifically, in assistance problems, an agent (the "assistant") aims to assist a teammate in solving a

given sequential task under uncertainty. In a closely related work, Ribeiro et al. [9] already consider ad hoc teamwork involving a human teammate. However, both works consider perfect observability and do not leverage the communicating capabilities of human users.

The two modules in our proposed architecture extend, on one hand, the decision-making process of Ribeiro et al. [9] to accommodate partial observability. On the other hand, the communication module enables our robot to leverage the communication capabilities of the human user towards the completion of the joint task.

Regarding Natural Language Processing (NLP), this is a field of computer science that enables machines to understand and process human communication [20] by transforming unstructured data, like audio or text, into structured data, which are more suitable for machines. It can also work in the opposite direction by generating communication for humans to understand easily.

Many works address the use of NLP in interaction with robots. Scheutz et al. [21] present the challenges of designing mechanisms that allow robots to develop human dialogues in interactions between humans and robots. In addition to building a small survey of the area, its main objective is to help build better, more flexible robotic architectures that can enable more natural language dialogues between humans and robots. Furthermore, the authors briefly propose DIARC, a Distributed, Integrated, Affective, Reflective, and Cognitive architecture that allows robotic systems to conduct human dialogues without providing much detail about the techniques involved in the process.

Briggs et al. [22] use pragmatic and dialogue-based mechanisms to understand typical human directives and create suitable responses. Specifically, utterances are used to represent the speech act classification, as well as the speaker, robot, and semantics analysis. Then, rules associate the utterances with a tuple containing the set of inferred beliefs based on the intended meaning of the utterance. For instance, a question inquiring whether some assertive is true or false. Finally, a dialogue-based mechanism handles and generates the responses based on expectations generated by the utterances. Experiments were then conducted to show the viability of the proposed methodology in identifying indirect speech acts and coverage of the utterance forms.

Li et al. [23] use natural language processing to infer human-given commands for robots, by using keyword extraction, visual object recognition, and similarity computation. Its main intent is to use visual semantic information to allow a robot deduce task intents, avoiding simple keywords that map predefined tasks explicitly. The proposed method uses rule matching and conditional random fields to analyze and extract information from the processed sentences.

Despite several works that use natural language processing in robots, none of them are tailored to the ad hoc teamwork scenario. Our work thus presents a novel contribution to the scientific literature, namely an architecture that combines decision making and natural language processing for human-robot collaboration within ad hoc teamwork settings.

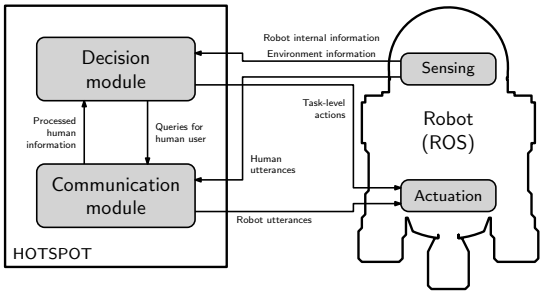


Fig. 1. Diagram depicting the interaction between the different modules in HOTSPOT.

III. The HOTSPOT Framework

This section presents the HOTSPOT framework for ad hoc teamwork involving humans and robots, which is the key contribution of this paper. Figure 1 presents the two main modules in the HOTSPOT architecture, namely:

- *A decision module* that is responsible for ad hoc teamwork decisions with the human. This module receives as input the robot and environment information from the sensors and the relevant information from the human speech (i.e., the information processed by the communication module). The decision module then uses such information to reconstruct/estimate the current state of the environment and the human-robot team. Finally, the robot uses the state information to estimate the current task (*task identification*), to identify how the human is executing such task (*teammate identification*), and to act accordingly (*planning*).
- *A communication module* that is responsible for the communication with the human user. It receives queries from the decision module and translates them into spoken utterances that the robot must execute (verbalize and animate). It is also responsible for processing human utterances, as perceived by the sensors, providing the decision module with their relevant information.

In the remainder of this section, we describe both modules in greater detail.

A. The Decision Module

The decision module is depicted in Fig. 2. The module processes the information coming from the sensors and communication module to estimate the *state* of the current task. Specifically, HOTSPOT maintains a distribution over possible states—a *belief*—which is updated from the perceived information using a standard Bayesian update.

The robot then uses information about possible tasks (stored in a *task library*) to infer the current task by checking which tasks in the library are most likely to yield the perceived information from the environment and the teammate. Finally, using the belief and task information, the robot plans the actions to complete the task. It also determines which (if any) communication actions it should perform towards the human.

In the continuation, we formalize each of these processes in detail.

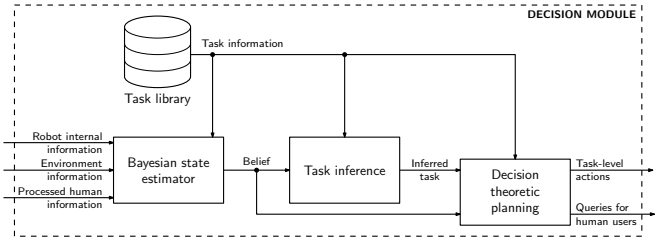


Fig. 2. Decision module in HOTSPOT.

Task description: We build on the works of Ribeiro et al. [9] and Melo and Sardinha [7] to formalize the possible tasks of the human-robot team using a decision-theoretic framework that accommodates the inherent uncertainty in scenarios involving robots in a principled manner.

We represent each possible task in the agent's library as a *multiagent Markov decision problem* (MMDP) [24], consisting of a tuple $\mathcal{M}_m = (\mathcal{X}, \mathcal{A}_0, \mathcal{A}_{-0}, \mathbf{P}_m, r_m, \gamma)$, where $\mathcal{X}$ is the set of all possible states, $\mathcal{A}_0$ is the set of actions available to the robot, A_{-0} is the set of actions available to the teammate (the human user), $\mathbf{P}_m$ describes the *dynamics* of the task m, r_m is a reward function, describing the *goal* of task m, and γ is a discount factor.

The states encode all task-relevant information, i.e., all information that, at each time step, the agents (the robot and the human) require to decide the next action. We write X_t to denote the state at time-step t, and $A_{0,t}$ and $A_{-0,t}$ to denote the actions of the robot and the human user at time step t, respectively. The state evolves according to the transition probabilities in $\mathbf{P}_m$, i.e., if T denotes the (unknown) current task,

$$\mathbf{P}_m(x' \mid x, a_0, a_{-0}) = \mathbb{P}\left[X_{t+1} = x' \mid X_t = x, A_{0,t} = a_0, A_{-0,t} = a_{-0}, T = m\right],$$

with $x, x' \in \mathcal{X}$, $a_0 \in \mathcal{A}$, and $a_{-0} \in \mathcal{A}_{-0}$. The transition probabilities describe the effect that the actions of the robot and the human user have on the *state*, given that the current task is m. Similarly, the reward function r_m encodes the goal of the human-robot team: the value $r_m(x, a_0, a_{-0})$ measures the *instant utility* of the robot executing action a_0 and the human executing action a_{-0} in state x, when the task is m.

Together, the human and the robot want to select their actions to maximize the total sum of rewards which, if $T = m$, can be written as

$$J = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_m(X_t, A_{0,t}, A_{-0,t})\right], \tag{1}$$

where $\gamma \in [0, 1)$ is a discount factor assigning greater value to rewards arriving earlier than those arriving later. Solving an MMDP thus consists of computing two individual *policies* for the two agents, π_0 and π_{-0}, each prescribing an action for each possible state and so that the prescribed actions jointly solve the MMDP—i.e., maximize the value in (1). Solving an MMDP can be done using standard dynamic programming techniques such as value or policy iteration [25].

In our setting, we consider that there is a set of M possible tasks, each described as an MMDP $\mathcal{M}_m = (\mathcal{X}, \mathcal{A}_0, \mathcal{A}_{-0}, \mathbf{P}_m, r_m, \gamma)$, where all tasks share the state and action spaces but may have different dynamics and goals. Furthermore, we assume that the robot does not know beforehand which is the task being performed—henceforth referred to as the *target task* T—but the human user does know. Task identification thus consists in inferring the target task from the information that the robot can observe during the interaction.

Bayesian state estimator: During the interaction, the robot can observe the information available through its sensors and information provided from the communication module regarding the human spoken utterances. We denote by Z_t the information observed by the agent—which we assume takes values in a finite set of possible observations, $\mathcal{Z}$. We denote by $\mathbf{O}$ the *observation probabilities*, which essentially provide a probabilistic description of the sensing process of the robot. In particular,

$$\mathbf{O}(z \mid x, a_0) = \mathbb{P}\left[Z_{t+1} = z \mid X_{t+1} = x, A_{0,t} = a_0\right],$$

with $z \in \mathcal{Z}$ and $x \in \mathcal{X}$. The observation probabilities describe how likely it is for the robot to observe z in state x, given that the last action of the robot was a_0.

Let $b_t(x)$ denote the probability that, at time step t, the state is $x \in \mathcal{X}$, given the history of observations and actions of the robot up to that time step (henceforth H_t). Let us further assume that, at time step t, the robot performs action a_0, and the human user performs action a_{-0}. As a consequence, the environment will transition to state X_{t+1} and the robot will observe $Z_{t+1} = z$. Then, if the target task is m, we can update our belief b_t using a standard Bayesian update to have

$$\begin{aligned} b_{t+1}(x) &= \mathbb{P}\left[X_{t+1} = x \mid H_{t+1}\right] \\ &= \frac{1}{\rho} \sum_{x' \in \mathcal{X}} \mathbf{P}_m(x \mid x', a_0, a_{-0}) \cdot \mathbf{O}(z \mid x', a_{-0}) b_t(x'), \end{aligned}$$

where ρ is a normalization constant.

There are two difficulties with using this update: first, we do not know which is the action of the human teammate, a_{-0}; and second, we do not know which is the target task, m.

To address the first difficulty, and since we assume that the teammate knows the target task, we consider that—if the target task is m—the action of the teammate can be *any* optimal action for the task m. Then, if we average out the action selection of our human teammate, we get the (task-dependent) transition probabilities

$$\bar{\mathbf{P}}_m(x' \mid x, a_0) = \frac{1}{\left|\mathcal{A}^*_{-0}(x)\right|} \sum_{a_{-0} \in \mathcal{A}^*_{-0}(x)} \mathbf{P}_m(x' \mid x, a_0, a_{-0}),$$

where $\mathcal{A}^*_{-0}(x)$ denotes the set of optimal teammate actions in state x. This yields a task-dependent belief update

$$\begin{aligned} b_{m,t+1}(x) &= \mathbb{P}\left[X_{t+1} = x \mid H_{t+1}, T = m\right] \\ &= \frac{1}{\rho} \sum_{x' \in \mathcal{X}} \bar{\mathbf{P}}_m(x \mid x', a_0) \mathbf{O}(z \mid x', a_{-0}) b_{m,t}(x'), \end{aligned} \tag{2}$$

where ρ is, again, a normalization constant.

Regarding the second difficulty, since the robot does not know beforehand the target task, it maintains a distribution p_t

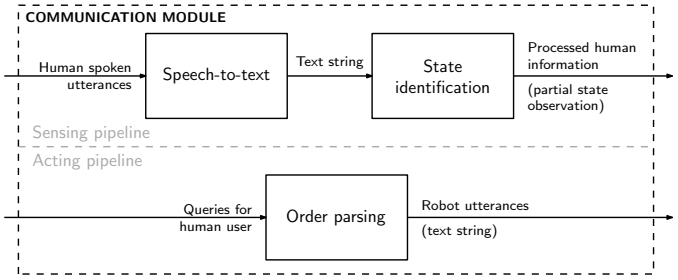


Fig. 3. The communication module, comprising both a *sensing* pipeline that converts speech to state information, and a *actuation* pipeline, converting communicating actions to text to be then spoken by the robot.

over the set of possible tasks. In other words, we write $p_t(m)$ to denote the probability that the target task is m given the history of observations and actions of the robot up to time step t. Then, given the distribution p_t, we can write the "average" belief at time step t as

$$b_t(x) = \sum_{m=1}^{M} p_t(m) b_{m,t}(x). \tag{3}$$

Task inference: We now describe how to maintain the distribution p_t, used to perform Bayesian state estimation. Much like with the state estimation, we adopt a Bayesian framework. Let T denote the unknown target task. Then, if the agent observed z at time step $t+1$ after executing a_0 at time step t,

$$\begin{aligned} p_{t+1}(m) &= \mathbb{P}\left[T = m \mid H_{t+1}\right] \\ &= \frac{1}{\rho}\mathbb{P}\left[Z_{t+1} = z \mid A_{0,t} = a_0, T = m, H_t\right] \cdot \\ &\qquad \mathbb{P}\left[A_{0,t} = a_0 \mid H_t\right] p_t(m), \end{aligned}$$

where, once again, ρ is the necessary normalization constant. We used the fact that the action selected by the agent at each moment depends only on the history of observations and not on the target task T. Then,

$$\begin{aligned} &\mathbb{P}\left[Z_{t+1} = z \mid A_{0,t} = a_0, T = m, H_t\right] \\ &\qquad = \sum_{x,x' \in \mathcal{X}} \mathbf{O}(z \mid x', a_0) \mathbf{P}_m(x' \mid x, a_0) b_{m,t}(x). \end{aligned}$$

Decision-theoretic planning: To decide what action to take, and given the uncertainty in the robot's perception of its state, we adopt as the planning approach a well-established information-gathering heuristic [26]. In particular, at each time step t, the robot selects its actions to balance *information gathering* and *task completion*.

Information gathering consists of selecting actions that decrease the uncertainty in the state estimation, b_t. Task completion actions are selected to solve the target task, T.

To this purpose, we compute the *normalized entropy* of b_t, given by

$$\bar{H}(b_t) = -\frac{1}{\log|\mathcal{X}|} \sum_{x \in \mathcal{X}} b_t(x) \log b_t(x).$$

The normalized entropy measures the uncertainty in the agent's belief. Let $b_{\max}(z, a_0)$ denote the robot's belief upon observing z after executing a_0 in task m from a belief with maximum entropy, i.e.,

$$b_{m,\max}(z, a_0) = \frac{1}{\rho} \sum_{x \in \mathcal{X}} \bar{\mathbf{P}}_m(x' \mid x, a_0) \mathbf{O}(z \mid x', a_0) \frac{1}{|\mathcal{X}|}.$$

We define the *information gain* associated with (z, a_0) as

$$\Delta H_m(z, a_0) = 1 - \bar{H}(b_{m,\max}(z, a_0)).$$

We also define the *reward gain* associated with (z, a_0) as the maximum reward that the robot can get upon observing z after executing a_0 in task m from a belief with maximum entropy, i.e.,

$$\begin{aligned} &\Delta R_m(z, a_0) \\ &\quad = \max_{a_0' \in \mathcal{A}_0} \sum_{x,x' \in \mathcal{X}} \bar{\mathbf{P}}_m(x' \mid x, a_0) \mathbf{O}(z \mid x', a_0) \frac{\bar{r}_m(x', a_0)}{|\mathcal{X}|}, \end{aligned}$$

with

$$\bar{r}_m(x, a_0) = \frac{1}{\left|\mathcal{A}^*_{-0}(x)\right|} \sum_{a_{-0} \in \mathcal{A}^*_{-0}(x)} r_m(x, a_0, a_{-0}).$$

Following Melo and Ribeiro [26], we define an *information gathering reward function* as

$$\begin{aligned} r_{m,\text{info}}(x, a) = \sum_{z \in \mathcal{Z}} &\mathbb{P}\left[Z_{t+1} = z \mid X_t = x, A_{0,t} = a_0, T = m\right] \\ &\cdot \Delta H_m(z, a_0) \cdot \Delta R_m(z, a_0). \end{aligned}$$

Then, for each task m, we can now define two standard MDPs, $(\mathcal{X}, \mathcal{A}_0, \bar{\mathbf{P}}_m, r_m, \gamma)$ and $(\mathcal{X}, \mathcal{A}_0, \bar{\mathbf{P}}_m, r_{m,\text{info}}, \gamma)$, which can be solved to yield two optimal Q-functions [25], Q^*_m and $Q^*_{m,\text{info}}$.

Finally, the action selected at each step t is given by

$$\begin{aligned} a_{0,t} = \operatorname{argmax} &\sum_{m=1}^{M} p_t(m) \sum_{x \in \mathcal{X}} b_t(x) \\ &\cdot \left[(1 - \bar{H}(b_t)) Q^*_m(x, a) + \bar{H}(b_t)) Q^*_{m,\text{info}}(x, a)\right]. \end{aligned}$$

When the uncertainty in b_t is high (close to 1), the robot selects an information gathering action, i.e., an action that maximizes $Q^*_{m,\text{info}}$; when the entropy is low (close to 0), the robot selects a task competing action, i.e., an action that maximizes $Q^*_m(x, a)$.

B. Communication Module

Figure 3 depicts the communication module, which plays two roles in the overall HOTSPOT architecture. First, it plays a *sensing role*, transforming the human speech (captured through a microphone) into state information that is then used by the decision module. Second, it also plays an *acting role*, converting the communication actions received from the decision module into utterances that the robot then speaks to the human user. Each role corresponds to a well-defined pipeline, as depicted in Fig. 3.

Concerning the sensing pipeline, the communication between the robot and the human user occurs through speech, captured by a microphone and transformed into audio data, usually in the form of a .wav file containing the recorded human spoken utterances. Next, the audio data is transcribed by

a speech-to-text block into a text format that better represents the audio in the language being spoken. Subsequently, in the state identification block, the transcribed text is passed into an NLP processor to extract semantic information, which is then translated into a partial state description.

The description of the state is then used as one of the several inputs to the decision module, which returns action(s) based on the behavior explained in Section III-A. These actions are mapped both into task-level and communication actions. The latter actions are passed to the communication module once again (i.e., the acting pipeline).

On the other hand, the acting pipeline is responsible for converting the communication actions into a text string, which is then sent to the robot. Specifically, this is done by the order parsing module, which provides the utterances to the robot, thus closing the cycle of communication between HOTSPOT and the robotic platform.

C. Interaction with the Robot.

The interaction between HOTSPOT and the robot relies on the *robot operating system* (ROS) [27], which is responsible for sensor handling (namely, processing all sensor readings), robot control, and communication. In other words, ROS supervises all sensing and actuation of the robot, and it is through a ROS interface that HOTSPOT interacts with the robot.

In particular, ROS collects all sensor data, arriving both from the robot sensors—such as odometry sensors used in dead-reckoning, lasers, contact sensors, etc—and environment sensors—such as microphones, cameras, and other sensors that may exist in the environment. The speech data is sent to the communication module, while the remaining sensor data is processed and sent to the decision module.

ROS is also responsible for the actuation of the robot. In particular, it receives the task-level actions (such as moving) from the decision module and the text strings (corresponding to the utterances that the robot should speak) from the communication module and performs these on the robot.

IV. Experimental setup

To evaluate the effectiveness of our approach, we created a controlled environment where a live robot interacts with a human teammate. In this environment, the robot aims to assist the human in cleaning up a room, with the interaction being restricted by a set of rules from the Toxic Waste Domain.

A. The Toxic Waste Domain

The Toxic Waste domain has a two-agent team composed of a human cleaner and a robot container. The team is in a building with several rooms and has to clean three rooms with toxic or radioactive waste. A specific task from this domain lays out the rooms in a topological map, where the nodes represent the rooms, and the edges represent the doors that connect them. In addition, some rooms may contain toxic material on the ground. Hence, in each time step, both the human and the robot may choose to move from one room (node) to another or stay in the same node (i.e., no-op).

Whenever it finds itself in a node containing toxic waste, the human can pick it up from the ground or release it (if he is already holding it). When the human picks up toxic waste, he must remain standing still on his current node and wait for the robot to get close to dispose of the toxic material into the robot's container. The robot also has an additional action to query the human by location (which the human may or may not respond to). Finally, in each time step, the robot receives its current location as an observation, inferred by the dead reckoning module.

Figure 4 shows the Toxic Waste domain that we created within our laboratory. Precisely, we recreate two tasks by dividing our laboratory room into five distinct areas, representing separate rooms: 0 - door, 1 - open space, 2 - robot station, 3 - single workbench, and 4 - the double workbench. To represent the toxic waste that the human can collect, we use three colored balls placed in three different nodes, as depicted in Fig. 5. The location of the three balls models each task; that is, there are two possible tasks, each with the three balls placed in three respective areas, as shown in Fig. 6.

To represent the human cleaner, we rely on people from a small focus group who have been told the goal in advance and know how to act according to the domain's rules, namely: i) they may only make one move at a time, ii) they may reply to the robot's questions, and iii) they may only move from one area to the another if they are connected. We rely on Astro (Fig. 4) to be the robot container. It is a robot from our laboratory capable of moving around the room and possessing a front recipient equipped with an RFID sensor to detect when a ball is placed inside the recipient. For each person of the focus group, we randomly chose a task from the two possible tasks, with the human starting in the area of their preference and the robot always starting at the door.

B. The Decision Module

We instantiate the decision module as a Python 3 ROS node running on a laptop (connected via wifi to a ROS master node running on the robot). The implementation of the module requires only a library of possible tasks, which are then used by our Bayesian state estimator, task inference, and decision-making algorithm.

To model the tasks in the Toxic Waste domain in our framework, we must specify the corresponding MMDPs, as well as the observation space and probabilities, describing the sensing process of the robot. Each MMDP is a tuple $(\mathcal{X}, \mathcal{A}_0, \mathcal{A}_{-0}, \mathbf{P}_m, r_m, \gamma)$ with distinct transition probabilities and reward functions. Together with the specification of the observation space and observation probabilities for the robot, they provide all the necessary information required by the decision-making module.

In our experiments,

- The state space $\mathcal{X}$ contains information regarding both agents' nodes and the status of the three toxic materials (i.e., on the ground, picked up, or disposed of). In particular, a state $x \in \mathcal{X}$ is a tuple $(n_r, n_h, w_1, w_2, w_3)$, where n_r and n_h represent the node of the robot and human, respectively, and w_i represents the status of the toxic waste material i.

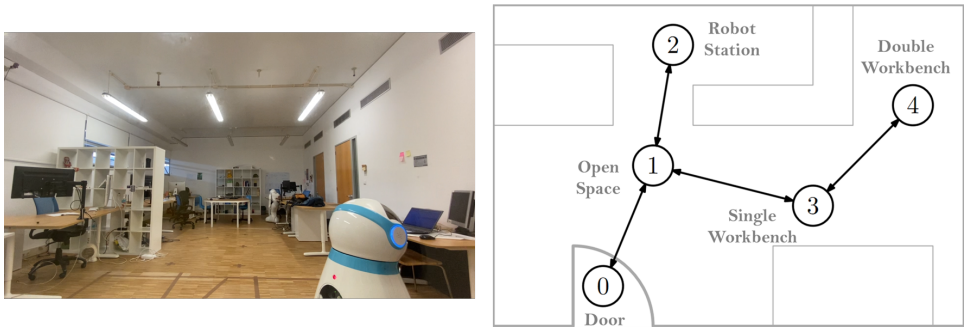


Fig. 4. Lab room used to simulate the Toxic Waste domain (left) and respective layout (right). Each area is represented as a node in a topological map.

Fig. 5. The three balls representing the toxic waste materials

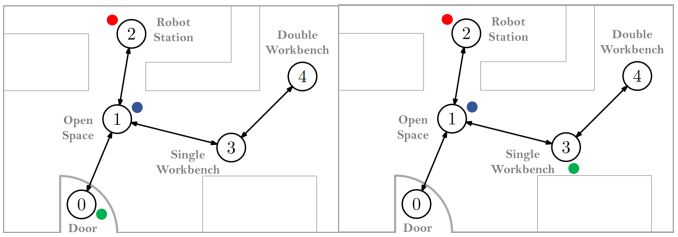


Fig. 6. The two task configurations (i.e., locations of the three balls representing the toxic waste materials)

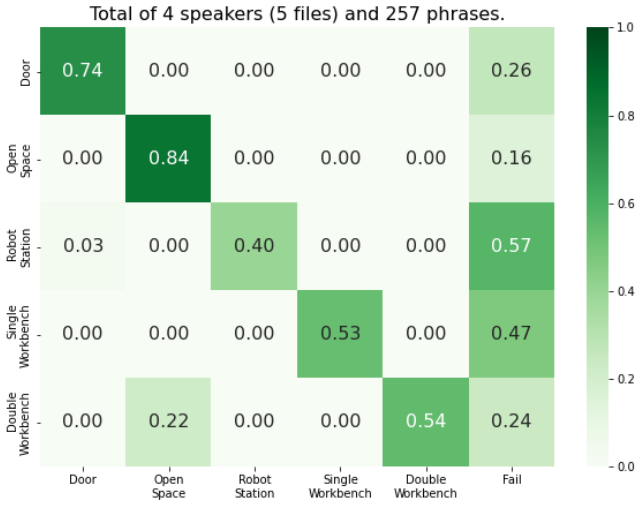


Fig. 7. Communication module's pipeline confusion matrix

- The action space for the robot, $\mathcal{A}_0$, has five possible actions available: *move the lowest-index node*, *move the second lowest-index node*, *move the third-index node*, *stand still*, and *ask the human for his location*. The human action space, $\mathcal{A}_{-0}$, has similar move actions and, additionally, a *pick waste* and *drop waste* actions.
- The transition probabilities $\mathbf{P}_m$ describe how the robot and human move as a consequence of their movement actions, and the status of the wast material as a consequence of the actions of the human user.
- The reward functions r_m assign a penalty of -1 for each toxic waste on the ground, -2 for each toxic waste on the human's hands, and 0 for each toxic waste material disposed.
- We use a discount $\gamma = 0.95$.
- The observations describe what the robot can observe regarding the state of environment and the human user. Specifically, each observation $z \in \mathcal{Z}$ corresponds to a tuple $(\hat{n}_r, \hat{n}_h, rfid)$, where $\hat{n}_r$ represents the robot's node (determined via dead reckoning) and $\hat{n}_h$ represents the human's node (determined from speech). *rfid* is a boolean flag indicating that the container detected the collection/disposal of a toxic waste.
- As for the observation probabilities $\mathbf{O}$, regarding the location of the robot and RFID sensor, we empirically assessed that the error in these measurements was negligible. As for the position of the human (perceived from human speech), we ran a preliminary study where, for each possible human node, we script out several phrases from a small focus group (i.e., 4 different speakers reading 257 different phrases). We then build a confusion matrix (Fig. 7) using Python's machine learning library scikit-learn, which tells, for each true node, the probability of identifying every other node or even failing to identify any node. Finally, to handle live errors, we smooth the probabilities when loading the model using the confusion matrix to ensure that no entry has an absolute zero.

C. The Communication Module

We implement the communication module as a Python 3 ROS node, and it runs on the same laptop as the decision module. In addition, we resort to the SpeechRecognition library[2] to convert human spoken utterances to text and break down the state identification component into two sub-components. The

[2]https://pypi.org/project/SpeechRecognition/

first sub-component is an entity recognition, which performs named entity recognition (NER) and syntactic parsing to extract the spoken location from the text string. The second sub-component is a node identifier, which takes the location found by NER and finds the correct node (represented by an integer) and the corresponding state. Finally, the node identification sub-component takes the location (string) outputted by NER and searches in a list of possible aliases for a string match. In other words, if an alias is found, its corresponding index is returned; otherwise, a symbolic value is then returned (-1).

Next, we describe each of the blocks in the communication module in further detail.

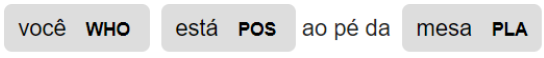
Speech-to-text: We use Google's speech-to-text API, packaged into a system's module, to transcript the human's inputs during the interactions with the robot. In these interactions, the human can send commands to the robot or, when asked, inform the robot's or his location. Therefore, the speech-to-text module is the first step in encoding a speech waveform into a useful representation of the human's inputs. The transcribed text, outputted by the module, is then sent to the state identification sub-module to extract semantic information, like the precise location of the involved actors.

State identification: The state identification sub-module uses NLP techniques, such as NER, to transform the transcribed text into locations from the robot's audio source. The primary responsibility of the NER is to extract the minimum information needed to obtain the location for the robot, which is the entity of the localization (the robot or the human) and the localization itself. These can be achieved through two possible analyses. The first one is by using a customized NER system with tags that indicate the important entities for the task. The second one is conducted through a syntactic parser of the transcribed text whenever the first approach cannot find any useful information.

Both approaches are implemented with Spacy [28], which is an open-source NLP Python library that uses Deep Learning techniques and statistical models, to implement NLP tasks for several languages, such as NER systems and shallow parsing.

The main advantage of Spacy is that, while having support for multiple pre-trained models, one can quickly build its model to evaluate particular entity tags. Spacy already has a built-in localization entity-tag set for NER that is not useful in our task because the target localizations are not the same as the ones trained by Spacy's module, like states, cities, and countries. Also, the entity in which the localization is being referred, the robot or human, may be described by pronouns (I, you, for instance) which are not classical named entities. Lastly, we want to build a relationship between tags, indicating that such localization *belongs* to a particular entity.

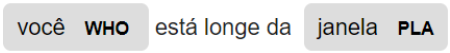
For instance, in the sentence, "*você está ao pé da mesa*", which means "you are by the table", in English, the word *você* (you) means the entity being referenced, in this case, the robot, the word *mesa* (table) indicates its position, and the verb *está* (are) is used here to indicate the relationship between the two entities. This same example can be expressed with its NER tags, as follows:



So, the newly created entity tags are:

- **WHO**, indicating who is being referenced;
- **PLA**, indicating the place being referenced;
- **POS**, indicating that this is indeed a valid relationship.

Note that one sentence may contain both WHO and PLA tags, **but** not a POS tag, which indicates that those entities do not necessarily correspond to a valid localization for this task. This is better illustrated with the following example:



In this example, which means "you are far from the window", no precise localization is informed. Nevertheless, we can use this example to train the other entities.

The idea behind training a customized NER system in Spacy is to feed its NLP pipeline with tagged examples of entities we want to build an extractor. Hence, we used examples from audio captures of three different Portuguese speakers and manually tagged them. We also added some counterexamples of positions that do not indicate a precise localization based on these examples. Table I presents a small extract of examples with their corresponding dictionaries used by Spacy.

TABLE I
SOME EXAMPLES FOR THE TRAINING OF THE NER EXTRACTOR

Text example	Entities dictionary
você está junto à janela	[(0, 4, 'WHO'), (18, 24, 'PLA'), (5, 9, 'POS')]
tu estás ao pé do Baxter	[(0, 2, 'WHO'), (18, 24, 'PLA'), (3, 8, 'POS')]
tu estás longe da bancada dupla	[(0, 2, 'WHO'), (17, 30, 'PLA')]

Our second approach is triggered whenever the system detects no relation, i.e., the NER extractor fails to find a full entity set. In this case, the system performs a full morphological and syntactic analysis, providing a full set of part-of-speech (POS) and universal dependencies (UD) tags [29] for the sentences being analyzed. We illustrate this situation with the example in Fig. 8, where all words and relationships from the sentence "you are by the door" (in English) are tagged. Note that the word "você" (you) is tagged as a pronoun (POS tag **PRON**) and the subject of the sentence (UD tag **nsubj**).

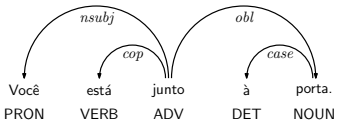


Fig. 8. Example of the morphological and syntactic analysis for the NER Module.

We can also see the analysis as a tree, as shown in Fig. 8, where, for instance, the root is the word *junto* ("by") since *está* ("is") is a copular verb[3] (UD tag **cop**). We can then traverse the tree to find the entities for the localization. In this case, the word *você* ("you") is the subject of the sentence, the word *está* ("are") is the copular verb, and, finally, the word *porta* ("door") is the object (UD tag **obj**) of the verb.

[3]A copular verb has the main function of joining the sentence's subject to its complement, like the verb "to be", for instance.

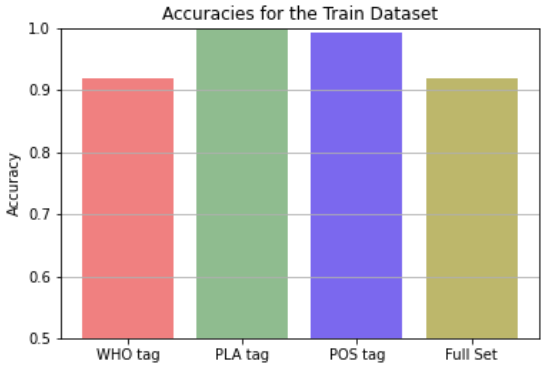


Fig. 9. Accuracies for all entities in the training data set.

By using both approaches in sequence (when needed), we achieve the performance depicted in Fig. 9, which shows that the module achieves excellent performance with the most common errors coming from the WHO tag, with an accuracy of 91.80%. This result is indeed true, mainly because there are a lot of possible examples in the training set that contains hidden subjects, such as: *estás ao pé da porta*. In this particular example, which means "(you) are by the door", although there is no explicit tag, one can evaluate that "who" is the robot. These hidden subjects are common, especially because they are usually answers to robot questions about a particular localization.

On the other hand, we achieve an accuracy of 100.0% when classifying $1,112$ entities with the PLA tag. Usually, places are easier to identify because of the use of prepositions right before them. Finally, we obtain an accuracy of 91.80% when classifying the whole aggregate (full entities set) with 528 examples in the dataset.

Order Parsing Module: Whenever the decision module returns an action identifier representing a query for the human user, the system feeds this identifier into a module called the order parsing module. Specifically, it has the responsibility of mapping each possible action to a given utterance. However, to avoid repetition, a bank of possible utterances is associated with each action. The module then outputs one of the utterances, which turns into an input to the robot's actuators (the text-to-speech engine connected to the speakers).

D. Metrics

Given that our approach has several independent modules, we evaluate the system with the following metrics: i) the number of steps it takes for a team to solve a task, ii) the accuracy in identifying the correct task, iii) the belief vector's entropy in each step, iv) the accuracy in recognizing the human's spoken utterances, and v) the accuracy in identifying the correct nodes.

E. Procedure

Given a focus group of 7 individuals, we performed a total of 9 independent trials. For every trial, we randomly pick a target task from the two possible, place the three balls on their respective nodes, and randomly select the robot's starting node. Then, we freely let the human teammate select its initial node. For each time step t, we run the interaction and record the following: i) the decision module's beliefs over all possible tasks, ii) the robot's action, iii) the human's action, iv) the human's spoken speech, v) whether the human's speech was informative enough to infer its node, vi) the speech recognition's output string, vii) the communication module's identified location, and viii) the observation z_t (comprised of the robot's node observation, inferred via dead-reckoning and the human's node observation, identified via speech-to-text and state identification). We also register the initial state x_0 containing the nodes of the two agents: the robot and the human.

To assess the introduced difficulties associated with live robotic experiments, we tested three agents in a simulated toxic waste environment, as follows: i) an agent following the same algorithm as the system used for the live setting; ii) an agent following the underlying MMDP's optimal policy, and iii) an agent following a random policy. We also recorded the number of steps it took to identify the target task in the simulated environment. Finally, all interactions start in the same initial states like the ones in the live setting.

V. Results

To run the experiments in a live environment with a human teammate and a robot, we explained how two agents interact, in a discrete-time environment, to each participant in the focus group. We also informed each participant beforehand that they could only execute one action at a time. However, two participants (out of seven) still performed two actions in some of the steps. Despite this problem, our approach overcame such obstacles and still identified the task being performed and assisted the human in completing the task.

Figure 10 shows the average number of steps our approach took to complete the task assigned to the team. It also plots the number of steps required for our approach in identifying the correct task from two possible tasks. In addition, we present the number of steps it takes for a baseline agent to run in a simulated environment, namely an agent following an optimal policy, representing the best possible performance.

From the results in Fig. 10, we observe that our approach always completed the target task in a near-optimal number of steps and quickly identified the target task using only partial observations (without observing the human's actions). As expected, the agent following the optimal policy solved the tasks in the fewest steps. We can also observe that, on average, our approach can identify the correct task quicker than an optimal policy. We can look deeper into the task identification by plotting the average entropy of the beliefs at each time step, as shown in Fig. 11.

We can first observe in Fig. 11 that the average entropy decreases with each passing time step. This is expected because the agent has more information to infer the correct task as the agent interacts with the environment. The second observation is that the average entropy does not reach 0.0, although it has dropped from 1.0 to almost 0.20. This result shows that our approach may end a trial without being 100% sure what the

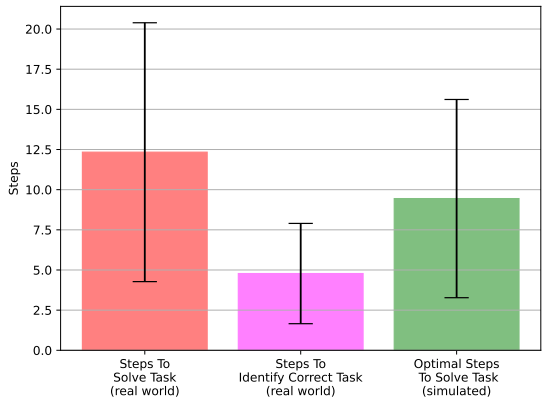


Fig. 10. The average number of steps to solve and to identify the target task. The baseline agent followed an optimal policy and ran in a simulated environment, with all simulated trials starting in the same initial states like the ones in the live trials.

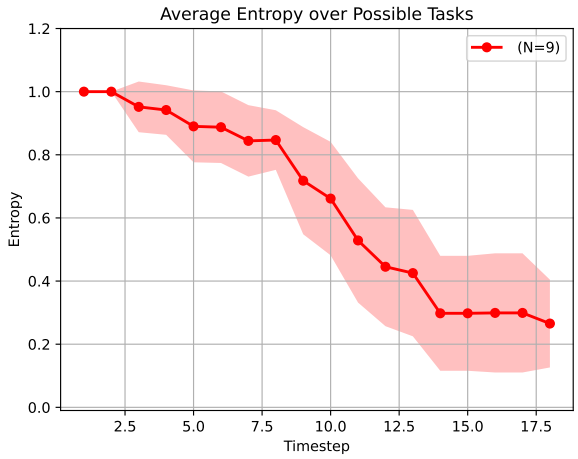


Fig. 11. Entropy of the beliefs at each time step, averaged over all nine trials

correct task is. However, this also indicates that our approach effectively solves the most likely task. Additionally, the fact that the average entropy is not 0.0 means that it may recover from task changes.

We also break down the evaluation of the NLP modules into three parts: i) the speech recognition module, ii) the NER module and iii) the node identification module. Then, having recorded all the human's spoken phrases plus the outputs of the three modules, in all time steps of the trials, we start reporting the accuracy of the speech recognition module. We also recorded, for each sentence, whether or not the sentence contained the information necessary to identify the human's location. Finally, from this set of informative phrases, we evaluate the accuracy of the NER module and, subsequently, of the node identification module. Fig. 12 plots the accuracies for these three modules in a live environment.

From Fig. 12, we observe that perfectly recognizing human speech is the hardest task of the three, with only a performance of 58.62%. From the spoken phrases that are informative enough to infer the correct human location, NER was able to identify the correct location in 77.77% of them. At last, the node identification module, which takes as input the NER location string, was able to correctly output the right human

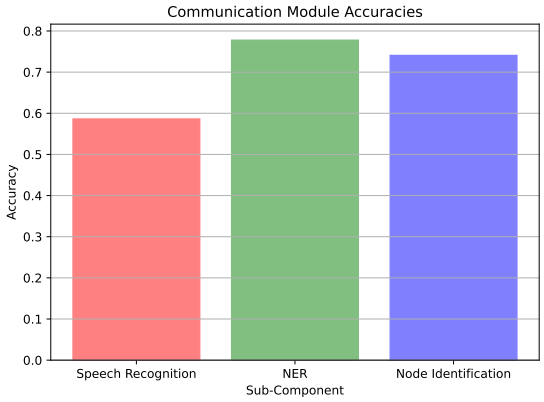


Fig. 12. The accuracies for the three NLP modules - Speech Recognition (58.62%), NER (77.77%) and Node Identification (74.07%).

nodes 74.07% of the time. Given that the goal of our approach was to work with partial observability, these accuracies show that our approach is effective and reliable when working with imperfect information.

VI. Conclusions and Future Work

Ad hoc teamwork addresses the decision-making problem of an agent when teamed to work with other unknown agents. Without any prior coordination or communication protocol, the agent must infer the cooperative task being performed, identify the teammates, and act to complete the task effectively.

In this work, we present HOTSPOT, a novel framework for ad hoc teamwork in human-robot teams. Specifically, our framework has two main modules, addressing the two key challenges in the interaction between a robot and a human teammate within ad hoc teamwork scenarios. The first module handles all the task-related decision-making challenges (i.e., task identification, teammate identification, and planning). The second module deals with the communication challenge between robots and humans by employing NLP techniques.

In order to evaluate our framework, we use a task that involves a mobile robot and a human teammate in a cooperative task of collecting objects in an open space, illustrating the main features of our framework in a real-world task. Results presented in Section V show that our approach always identified and completed the task with a near-optimal number of steps while using partial and imperfect information. We also observe that, on average, the proposed approach identified the task faster than the optimal policy, showing the potential that this approach has in a real task environment.

Although our approach has shown excellent results, we can always incorporate enhancements to further improve the proposed methods' performance. For example, one of the worst-performing components is the speech recognizer, which currently uses an online recognition service. Such service is not customized for the restricted vocabulary used in the human-robot conversation and is not suitable for the noisy environment of human-robot interactions. Moreover, besides the low average performance, any instability in the robot's Internet connection makes its use unfeasible. In this sense, developing an offline system customized for our domain,

would bring enormous advantages, both for the classifier performance and the response speed of the decision module.

Also, we plan to invest in other types of sensors for the robot, especially those that capture 360-degree scenes. That way, independent of the robot's position and orientation, we can apply computer vision techniques to enhance the robot's observation regarding the environment.

REFERENCES

[1] J. Pepito and R. Locsin, "Can nurses remain relevant in a technologically advanced future?" *Int. J. Nursing Sciences*, vol. 6, no. 1, pp. 106–110, 2019.

[2] P. Alves-Oliveira, S. Petisca, F. Correia, N. Maia, and A. Paiva, "Social robots for older adults: Framework of activities for aging in place with robots," in *Proc. 7th Int. Conf. Social Robotics*, 2015, pp. 11–20.

[3] F. Pratticò and F. Lamberti, "Mixed-reality robotic games: Design guidelines for effective entertainment with consumer robots," *IEEE Consumer Electronics Magazine*, vol. 10, no. 1, pp. 6–16, 2021.

[4] X. Miao, H. Lee, and B. Kang, "Multi-cleaning robots using cleaning distribution method based on map decomposition in large environments," *IEEE Access*, vol. 8, pp. 97 873–97 889, 2020.

[5] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on human-robot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics*, vol. 55, pp. 248–266, 2018.

[6] P. Stone, G. Kaminka, S. Kraus, and J. Rosenschein, "Ad hoc autonomous agent teams: Collaboration without pre-coordination," in *Proc. 24th AAAI Conf. Artificial Intelligence*, 2010, pp. 1504–1509.

[7] F. Melo and A. Sardinha, "Ad hoc teamwork by learning teammates' task," *Autonomous Agents and Multi-Agent Systems*, vol. 30, no. 2, pp. 175–219, 2016.

[8] S. Barrett, A. Rosenfeld, S. Kraus, and P. Stone, "Making friends on the fly: Cooperating with new teammates," *Artificial Intelligence*, vol. 242, pp. 132–171, 2017.

[9] J. Ribeiro, M. Faria, A. Sardinha, and F. Melo, "Helping people on the fly: Ad hoc teamwork for human-robot teams," in *Proc. 20th EPIA Conf. Artificial Intelligence*, 2021, pp. 635–647.

[10] S. Barrett, "Making friends on the fly: Advances in ad hoc teamwork," Ph.D. dissertation, The University of Texas at Austin, December 2014.

[11] S. Barrett and P. Stone, "Cooperating with unknown teammates in complex domains: A robot soccer case study of ad hoc teamwork," *Proc. 29th AAAI Conf. Artificial Intelligence*, pp. 2010–2016, 2015.

[12] P. Stone and S. Kraus, "To teach or not to teach?: Decision-making under uncertainty in ad hoc teams," in *Proc. 9th Int. Conf. Autonomous Agents and Multiagent Systems*, 2010, pp. 117–124.

[13] P. Stone, G. Kaminka, and J. Rosenschein, "Leading a best-response teammate in an ad hoc team," in *Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets*, ser. Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, 2010, vol. 59, pp. 132–146.

[14] N. Agmon and P. Stone, "Leading ad hoc agents in joint action settings with multiple teammates," in *Proc. 11th Int. Conf. Autonomous Agents and Multiagent Systems*, 2012, pp. 341–348.

[15] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning and Games*. Cambridge University Press, 2006.

[16] M. Spaan and N. Vlassis, "PERSEUS: Randomized point-based value iteration for POMDPs," *J. Artificial Intelligence Res.*, vol. 24, pp. 195–220, 2005.

[17] K. Genter, T. Laue, and P. Stone, "Three years of the RoboCup standard platform league drop-in player competition," *J. Autonomous Agents and Multi-Agent Systems*, vol. 31, no. 4, pp. 790–820, 2017.

[18] P. MacAlpine, K. Genter, S. Barrett, and P. Stone, "The RoboCup 2013 Drop-in Player Challenges: Experiments in ad hoc teamwork," in *Proc. 2014 IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2014, pp. 382–387.

[19] A. Fern, S. Natarajan, K. Judah, and P. Tadepalli, "A decision-theoretic model of assistance," in *Proc. 20th Int. Joint Conf. Artificial Intelligence*, 2007, pp. 1879–1884.

[20] C. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[21] M. Scheutz, R. Cantrell, and P. Schermerhorn, "Toward human-like task-based dialogue processing for human robot interaction," *AI Magazine*, vol. 32, no. 4, pp. 77–84, 2011.

[22] G. Briggs, T. Williams, and M. Scheutz, "Enabling robots to understand indirect speech acts in task-based interactions," *J. Human-Robot Interaction*, vol. 6, pp. 64–94, 2017.

[23] Z. Li, Y. Mu, Z. Sun, S. Song, J. Su, and J. Zhang, "Intention understanding in human-robot interaction based on visual-NLP semantics," *Frontiers in Neurorobotics*, vol. 14, no. 610139, 2021.

[24] C. Boutilier, "Planning, learning and coordination in multiagent decision processes," in *Proc. 6th Conf. Theoretical Aspects of Rationality and Knowledge*, 1996, pp. 195–210.

[25] M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience, 2005.

[26] F. Melo and M. Ribeiro, "Transition entropy in partially observable Markov decision processes," in *Proc. 9th Int. Conf. Intelligent Autonomous Systems*, 2006, pp. 282–289.

[27] Stanford Artificial Intelligence Laboratory, "Robotic Operating System." [Online]. Available: https://www.ros.org

[28] I. Montani *et al.*, "spaCy: Industrial-strength natural language processing in Python." [Online]. Available: https://doi.org/10.5281/zenodo.5517375

[29] M. de Marneffe, T. Dozat, N. Silveira, K. Haverinen, F. Ginter, J. Nivre, and C. Manning, "Universal Stanford dependencies: A cross-linguistic typology," in *Proc. 9th Int. Conf. Language Resources and Evaluation*, 2014.