Multiparty Verbal Interaction Between Humans and Artificial Agents

Lucrezia Grassi[1], Carmine Tommaso Recchiuto[1] and Antonio Sgorbissa[1]

[1] *University of Genoa, Via All'Opera Pia 13, 16145, Genoa, Italy*

Abstract

The study of verbal interaction between multiple humans and robots is an almost unexplored research field. This kind of interaction has been primarily analyzed in the literature focusing on cooperation to achieve a common task or on more technical aspects such as active speaker recognition. The presented work proposes a holistic approach to solve the problem: a cloud architecture that allows social robots and artificial agents to interact verbally with a group of people. The system can recognize the active speaker and decide who to address based on the developed policies while also correctly keeping track of the conversation state.

Keywords

Autonomous Conversation, Multiparty Interaction, Human-Robot Interaction, Social Robotics

1. Introduction

Social Robotics aims to develop robots that can provide physical and cognitive support in a socially interactive way. During the interaction, one of the main issues is the knowledge acquisition problem. To solve this problem, the system should have the capability of learning through interaction. The agent should recognize new relevant information, update its knowledge appropriately, and use new information to adapt its behavior when interacting with the user. This problem, which is already very challenging, is made more complex when a social agent communicates with multiple people simultaneously. In this context, the robot should not only be able to acquire knowledge but also to recognize its interlocutors to correctly associate relevant information with the person it relates to. Moreover, such a system should emulate the conversation patterns that typically emerge when more humans interact with each other [1], a problem that has almost been ignored in the Social Robotics literature where human-robot interaction is typically one-on-one. Among the problems to be addressed, the system will have to keep track of the conversation state with different users and recognize who is talking to provide the most appropriate response [2].

Currently, there are few robots capable of autonomously interacting with multiple users at the same time, although this type of interaction frequently occurs for humans. In a multiparty spoken dialogue system, such as the one described in [3], the agent can discriminate between

9th Italian Workshop on Artificial Intelligence and Robotics (AIRO 2022)

lucrezia.grassi@edu.unige.it (L. Grassi); carmine.recchiuto@dibris.unige.it (C. T. Recchiuto); antonio.sgorbissa@unige.it (A. Sgorbissa)

0000-0001-6363-3962 (L. Grassi); 0000-0001-9550-3740 (C. T. Recchiuto); 0000-0001-7789-4311 (A. Sgorbissa)

CEUR Workshop Proceedings (CEUR-WS.org)

multiple users using the information provided by a Kinect. However, the agent's conversational capabilities are very basic and limited, and long-term engaging and natural conversation between multiple parties is still an open problem. In addition, the agent is not able to engage multiple users simultaneously in the conversation, but only one at a time. The tracking and fusion aspects of multiparty interactions with artificial agents are studied in [4], where a system with a life-size virtual agent and a social robot is introduced. The system focuses only on a user entry/exit mechanism with re-identification of users, but not on the conversation, and can currently keep accurate track of only two users. The literature suggests that voice plays an important role when trying to determine who is the talking person: several techniques for speech recognition have already been studied [5] and some approaches work even in noisy and unconstrained conditions [6].

On the other side, the dynamics existing in group conversations have been deeply examined by researchers in the field of psychology. Several studies have found that increasing the number of people in a conversation creates systematic challenges for speakers and listeners, a phenomenon that is called "the many minds problem" [1]. Specifically, when more people interact with each other, the basic mechanisms of a conversation are altered, such as "turn-taking" (i.e., a type of organization in conversation in which participants speak one at a time in alternating turns), "floor-time" or "air-time" (i.e., the time participants use to speak), and the type of feedback listeners provide to the speaker.

The presented work:

- introduces a software architecture to empower a robot with the capability of recognizing the users participating in a conversation;
- implements different strategies to control the dynamics of a group conversation, deciding which speaker to address, based on data gathered during the interaction.

Section 2 briefly describes the architecture of the cloud system. Section 3 presents the results of the experiments performed to assess the performance of the system in terms of average response time.

2. A Cloud Architecture for Multiparty Verbal Interaction

The cloud system for multiparty interaction has been developed starting from CAIR, a cloud software architecture developed for autonomous conversation [7], [8], [9]. In brief, CAIR is composed of two web services: the Dialogue Manager service which manages the dialogue and analyzes the user sentence to recognize the intention of talking about a specific topic, and the Plan Manager service which recognizes the intention of the user to make the agent execute a specific action. To provide appropriate answers and plans, the server exploits an Ontology containing all the topics, keywords, sentences, and plans used during the interaction with the user, as described in [10], [11]. These components may be observed in Figure 1, embedded in the server. A client (i.e., the software controlling the robot) can perform requests to the server using REST APIs, by providing to the cloud server the sentence pronounced by the user along with information about the status of the conversation [7].

The "red" elements in Figure 1 are the ones that allow an effective multiparty interaction. All the requests that arrive to the cloud are managed by the Hub service, which oversees forwarding

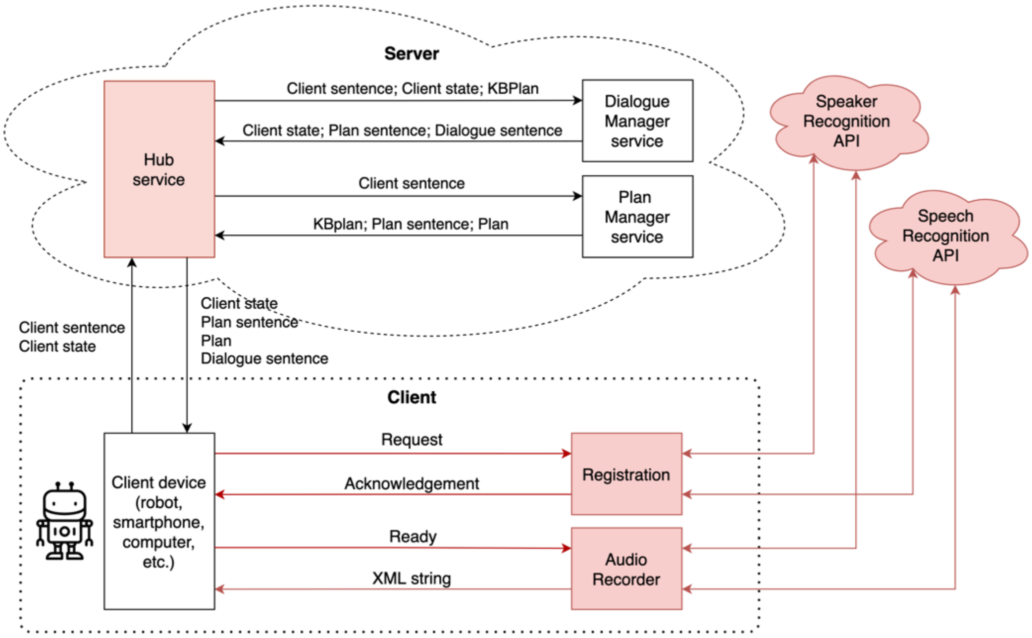

Figure 1: CAIR system architecture

them to the Dialogue Manager and the Plan Manager services. Moreover, the client has been expanded with two new services: Registration and Audio Recorder services. The Registration service is called every time a registration procedure is started. Suppose a new user wants to be recognized by the system: they simply have to trigger the registration procedure, at any moment of the conversation, by saying a sentence such as “Registration" or “Learn my voice" and the system will associate a new profile ID to the user, who will be asked to provide their name and gender and to talk for 20 seconds to complete the enrolment. The Audio Recorder service starts acquiring the audio when the Root Mean Square (RMS) of noise exceeds a certain threshold. This service sends the audio pieces to the Speech Recognition API to obtain the transcription, and to the Speaker Recognition API to obtain the ID of the corresponding speaker (if registered). After a final silence exceeding a certain threshold, the service returns an XML string containing the transcribed pieces of text, each tagged with the ID of the corresponding speaker. Eventually, the client sends the string to the Hub, along with the client state. Let us specify that the exchange of messages between the client and the Hub proceeds until one of the users decides to terminate the interaction by saying a predefined sentence such as “Goodbye" or “Disconnect".

Also, the client state has been expanded as it now contains statistics related to the speakers, such as a matrix containing the probability that a speaker talks after another, a matrix with the number of times a speaker talked after another in the same or successive turn, the total number of turns of each user, the average topic distance between speakers, the a priori probability that a speaker talks, a moving window keeping track of the turns, and other information. The moving window, stored on the client device, is a fundamental element of the state: it contains information about the conversation turns of the last M "active" minutes. For each turn, the moving window stores the ID of the speaker, the speaking time, and the number of words said. If the sum of the speaking times of the turns in the moving window exceeds M, the first turn is removed and the latest one is added (FIFO queue).

The information contained in the moving window has been used to develop two control policies, based on the analysis of group dynamics: the "dominant" policy and the "submissive" policy. The policies are implemented as functions that take as input the data contained in the moving window and output the speaker to address. The first policy recognizes and addresses the dominant user among the group of people interacting with the robot, while the second one recognizes and addresses the user who participates less in the conversation (submissive). Participation in the conversation is measured through a weight D_i that accounts for both speaking time and number of words, as they turned out to be the most relevant indicators to detect dominance [12]. To compute D_i for each speaker S_i, the percentage of their speaking time (T_i) and the number of words (W_i) in the moving window should first be measured. Then, D_i is computed as:

$$D_i = \gamma_1 T_i + \gamma_2 W_i, \tag{1}$$

where γ_1 and γ_2 are two gains that indicate how much importance is given to the speaking time and to the number of words when determining the dominance. The addressed speaker when applying the "dominant" policy will be:

$$S_i[M] \qquad where \qquad M = argmax_i(D_i) \tag{2}$$

while the addressed speaker when applying the "submissive" policy will be:

$$S_i[m] \qquad where \qquad m = argmin_i(D_i). \tag{3}$$

The third policy developed is the "community" policy. This policy is based on the idea that it is possible to identify sub-groups (i.e., communities) among the people in a group. To identify the communities, we use a matrix containing the probability that a speaker talks after another. Such a matrix is transformed into an undirected graph where the nodes represent the speakers, and the probabilities are the weights of the edges. The Louvain algorithm is then applied to the weighted graph to obtain the best partition of the nodes in communities [13]. The algorithm starts from a singleton partition in which each node is in its community. Then, it moves individual nodes from one community to another to find a partition. Based on this partition, the algorithm creates an aggregate network and moves individual nodes in such a network. These steps are repeated until the quality cannot be further increased. Once the best partition has been obtained, the result of the algorithm is used by the policy to address a random speaker of a different community at every turn. The policy aims to control the conversation by always maintaining a singleton partition and avoiding having speakers divided into sub-groups. Let us specify that the policy that the system should use can be chosen before starting the interaction.

3. Preliminary Results

An example of the interaction of multiple users with the described system may be observed in this video[1]. Moreover, an experimental protocol to assess the impact of the developed policies has already been defined and approved by the ethical committee. Participants will be divided into four groups (a control group and three experimental groups) and they will have to participate in a conversation with the robot. During the experiments, the robot will assume the role of a "moderator" applying the developed policies. Data gathered during the experiments will allow us to determine how the different policies impact participants' perceptions of the robot and the overall quality of the conversation.

Also, preliminary tests have been carried out to assess the capability of the system to deal with multiple client devices simultaneously. In particular, we performed the Baseline test to evaluate the performance of the system in terms of average response time (i.e., the difference between the time when the request was sent by the client and the time when the response was fully received). To do this, we considered four different payloads, each containing the client sentence and a dialogue client state of different sizes (from empty to full), to understand the impact of the request data size on the response time. For each of these scenarios, 30 requests spaced five seconds apart were performed by a single thread/client. From the results reported in Figure 2, it can be observed that, even with the maximum payload, the average response time is still very low (within 200 ms).

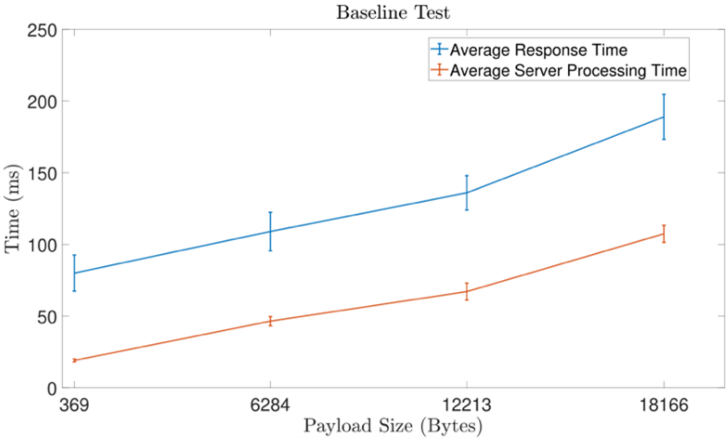

Figure 2: Results of the Baseline test with the 4 payload sizes.

As the objective is to empower a variety of devices with the ability to hold a long-term conversation with one or more users, it is fundamental that the system can manage contemporary connections from a growing number of clients. For this reason, we also performed the Scalability test to assess how the average response time increases with a growing number of requests. The test was carried out by simulating an increasing number of N users performing requests simultaneously, using the greatest request payload. The established threshold for these experiments is one second, which is below the delay reported in experiments about people's

[1] https://www.youtube.com/watch?v=TpCGqFZLN4k

perception during a dialogue with a conversational system [14, 15]. Setting a lower threshold arranges for variations that can be due to the load, the network performance, or the additional time required to perform the speech-to-text transcription. This ensures higher satisfaction during the conversation. Keeping this in mind, the results of the Scalability test, shown in Figure 3, revealed that the system can support up to 20 simultaneous requests without negatively affecting the user's perception.

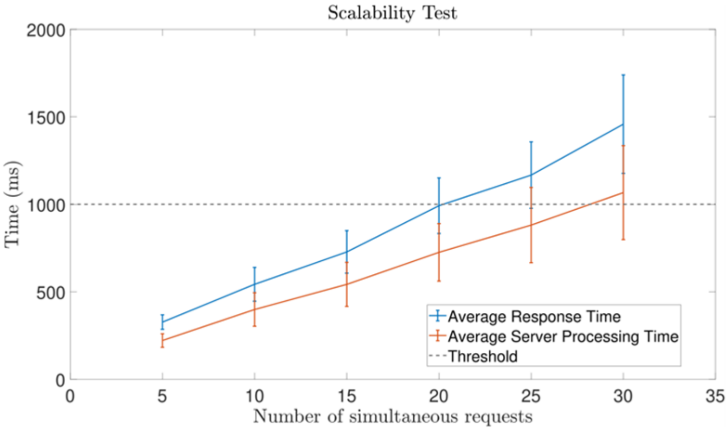

Figure 3: Results of the Scalability test with a growing number of simultaneous requests.

4. Conclusion

The paper presented the architecture of a cloud system allowing robots and other devices to verbally interact with multiple people simultaneously. The work also presented and discussed the results of experiments aimed at assessing the performance of the system in terms of average response speed. These preliminary findings provided us with the basis to size the system, paving the way to a sustainable solution for verbal interaction with low-cost robots and other intelligent devices.

References

[1] G. Cooney, A. M. Mastroianni, N. Abi-Esber, A. W. Brooks, The many minds problem: disclosure in dyadic versus group conversation, Current Opinion in Psychology 31 (2020) 22–27.

[2] T. Kinnunen, H. Li, An overview of text-independent speaker recognition: From features to supervectors, Speech communication 52 (2010) 12–40.

[3] A. Pappu, M. Sun, S. Sridharan, A. Rudnicky, Situated multiparty interaction between humans and agents, in: International Conference on Human-Computer Interaction, Springer, 2013, pp. 107–116.

[4] Z. Yumak, J. Ren, N. M. Thalmann, J. Yuan, Tracking and fusion for multiparty interaction

with a virtual character and a social robot, in: SIGGRAPH Asia 2014 Autonomous Virtual Humans and Social Robot for Telepresence, 2014, pp. 1–7.

[5] J. P. Campbell, Speaker recognition: A tutorial, Proceedings of the IEEE 85 (1997) 1437–1462.

[6] T. Kinnunen, H. Li, An overview of text-independent speaker recognition: From features to supervectors, Speech communication 52 (2010) 12–40.

[7] L. Grassi, C. T. Recchiuto, A. Sgorbissa, Cloud services for social robots and artificial agents, The 8th Italian Workshop on Artificial Intelligence and Robotics - AIRO 2021 (2021).

[8] L. Grassi, C. T. Recchiuto, A. Sgorbissa, Sustainable verbal and non-verbal human-robot interaction through cloud services, arXiv preprint arXiv:2203.02606 (2022).

[9] C. Recchiuto, L. Gava, L. Grassi, A. Grillo, M. Lagomarsino, D. Lanza, Z. Liu, C. Papadopoulos, I. Papadopoulos, A. Scalmato, et al., Cloud services for culture aware conversation: Socially assistive robots and virtual assistants, in: 2020 17th International Conference on Ubiquitous Robots (UR), IEEE, 2020, pp. 270–277.

[10] C. T. Recchiuto, A. Sgorbissa, A feasibility study of culture-aware cloud services for conversational robots, IEEE Robotics and Automation Letters 5 (2020) 6559–6566.

[11] L. Grassi, C. T. Recchiuto, A. Sgorbissa, Knowledge-grounded dialogue flow management for social robots and conversational agents, International Journal of Social Robotics (2022) 1–21.

[12] M. S. Mast, Dominance as expressed and inferred through speaking time: A meta-analysis, Human Communication Research 28 (2002) 420–450.

[13] X. Que, F. Checconi, F. Petrini, J. A. Gunnels, Scalable community detection with the louvain algorithm, in: 2015 IEEE International Parallel and Distributed Processing Symposium, IEEE, 2015, pp. 28–37.

[14] Z. Peng, K. Mo, X. Zhu, J. Chen, Z. Chen, Q. Xu, X. Ma, Understanding user perceptions of robot's delay, voice quality-speed trade-off and gui during conversation, in: Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems, CHI EA '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 1–8.

[15] T. Shiwa, T. Kanda, M. Imai, H. Ishiguro, N. Hagita, How quickly should communication robots respond?, in: HRI 2008, 2008, pp. 153–160.