DEVELOPMENTAL COLLABORATIVE INTELLIGENCE FOR EMBODIED ROBOTIC AGENTS

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN THE FACULTY OF SCIENCE AND ENGINEERING

2021

Samuele Vinanzi

Department of Computer Science
Contents

Abstract 13
Declaration 14
Copyright 15
Acknowledgements 16

1 Introduction 18
  1.1 Motivations 18
  1.2 Research Questions 20
  1.3 Aims and Objectives 20
  1.4 Contributions to Knowledge 21
  1.5 Publications 22
  1.6 Thesis Structure 23

2 Background 25
  2.1 Introduction 25
  2.2 Collaborative Intelligence 26
    2.2.1 Cognitive Foundations 26
    2.2.2 Computational Models 27
  2.3 Intention Reading 29
    2.3.1 Cognitive Foundations 29
    2.3.2 Computational Models 31
  2.4 Trust and Theory of Mind 32
    2.4.1 Cognitive Foundations 32
    2.4.2 Computational Models 34
  2.5 Developmental Robotics 35
3 Intention Reading

3.1 Introduction .................................................. 40
3.2 Methods ...................................................... 41
  3.2.1 Clustering ................................................. 41
    3.2.1.1 K-Means ........................................ 42
    3.2.1.2 X-Means ........................................ 43
    3.2.1.3 Cluster Evaluation: Silhouette Analysis .......... 43
  3.2.2 Principal Component Analysis ......................... 44
  3.2.3 Hidden semi-Markov Models ........................... 45
3.3 Model Description ........................................... 47
  3.3.1 Low-Level Skeleton Clustering and Action Representation ...... 48
    3.3.1.1 Skeleton Generation ............................ 48
    3.3.1.2 Dynamical Clustering ............................ 49
  3.3.2 High-Level Goal Probabilistic Inference ................. 51
  3.3.3 Robot Control ....................................... 54
3.4 Experiment ................................................. 54
  3.4.1 Experimental Setup .................................... 54
  3.4.2 Procedure ........................................... 54
3.5 Results and Discussion ..................................... 56
3.6 Conclusions ............................................... 58

4 The Role of Social Cues for Goal Disambiguation ............ 61
  4.1 Introduction ............................................... 61
  4.2 Methods ................................................... 62
    4.2.1 Multimodal Machine Learning ....................... 62
    4.2.2 Bayesian Networks .................................. 63
    4.2.2.1 Parameter learning ............................. 65
  4.3 Model Description ........................................ 66
    4.3.1 Feature-Space Split Clustering ..................... 66
List of Tables

3.1 Trained experimental intentions: each goal is associated to a sequence of cluster transitions. ............................................. 56
3.2 Experimental results for the intention reading experiment. See Section 3.5 for the full explanation. .......................... 58
4.1 Action encodings for the experiment on goal disambiguation. Each goal is associated to a sequence of cluster transitions. .......................... 76
4.2 Experimental results for the goal disambiguation experiment. See Section 4.5 for the full explanation. ...................... 77
5.1 Bayesian network node values for both a Mature ToM (1) and an Immature ToM (2) agent. The rows of the tables represent the posterior probability distributions associated with each node of the networks, for both decision making (DM) and belief estimation (BE) tasks. In each setting, the informant is suggesting position a. ..................... 97
6.1 Mean and standard deviation of the success rates calculated on the interactions performed by the stochastic simulated humans. .................. 111
# List of Figures

2.1 The humanoid robots used in this research. Sources: Italian Institute of Technology, Rethink Robotics, SoftBank Robotics.  
37

3.1 Basic structure of a HSMM. The hidden states (the goals) produce a sequence of visible observations (the human’s actions). Each state is moreover associated with a duration probability, which directly reflects the number of observations it will emit.  
47

3.2 Overview of the intention reading architecture. The low-level extracts skeleton data from the optical stream, forms clusters and uses them to represent actions as transitions through clusters. The high-level uses this encoding to probabilistically infer the goal pursued by the partner. The robot control is in charge of performing the collaborative behavior.  
48

3.3 A comparison between the skeletal keypoints extracted from the camera image (a) and the reduced keypoint set computed by the system (b).  
49

3.4 A diagram showing the training process of the low-level module. The image acquired by the robot’s eye camera is used to generate a keypoint representation of the human, which is then normalized before it can be clustered.  
50

3.5 Skeleton clustering. The observed skeletons are projected onto their 2 principal components $\alpha$ and $\beta$ and partitioned in similar postures. An action is defined as a particular transition, for example the goal “wave” is performed by a sequence of body postures that transition from cluster 3 to cluster 1 to cluster 2, hence the encoding [3, 1, 2].  
51

3.6 Experimental setup for the block building game during the pilot experiment of this line of research.  
55
3.7 An example of the four structures taken into account in the intention reading block placing experiment. In this case, the color combination chosen was: $\gamma_1 = green \land \gamma_2 = blue$. 

3.8 The 2D projection of the resulting clustering obtained during the training phase of the block building game. Skeletons have been divided into three groups of similar postures and the numbered dots represent the centroids. Each data point shows the bounding box of the human partner as seen from the robot’s eyes, the position of which was fixed on the opposite side of the table.

3.9 Decomposition of one of the intentions that are trained to the robot during the block building game, corresponding to the goal “wall”. The skeletons showed are the ones that triggered a cluster transition and show the visible upper body of the demonstrator manipulating blocks on the table.

4.1 An example of Bayesian network with two nodes representing boolean random variables. The tables shown next to them are their respective conditional probabilities tables. Note how the size of the latter depends on the number of the parents of that node.

4.2 A visual representation of the computation performed by FSSC. The samples at level 0 are clustered in the feature-space $f_1$ and produce several groupings at level 1. These are once again separately clustered in the feature-space $f_2$. The leaf nodes (in this example, at level 2) represent the final clusters.

4.3 The Bayesian network used for high-level probabilistic goal prediction. The top node represents the intention of the observed partner, whilst the bottom node symbolize the observations (the action encoding symbols produced by the low-level module).

4.4 The Sawyer robot used in the intention reading and goal disambiguation experiment involving an interactive block placing game.

4.5 Experimental setup for the block building game. (a) Schematic of the playing table, depicting the position of the 4 colored blocks: blue (B), orange (O), red (R) and green (G). (b) The 8 admissible block sequences obtained by picking blocks alternatively from each side. These sequences are the goals for this scenario.
4.6 An example of the post-processing operation which is performed on the training set. Suppose we are considering only 4 out of the 8 goals, the aim is to correct any irregularities in the data. (a) A possible partial training set that associates a goal label with a sequence of cluster IDs. The last value of GORB is incorrect due to a noisy observation. (b) The statistical mode for the end position B is calculated across the partial dataset: 3 out of 4 entries have assigned it to cluster ID 3. (c) The correct value is substituted in the dataset, fixing the error and allowing a correct training of the system.

4.7 Clustering results. (a) First level of FSSC in $f_1$ feature-space (skeleton keypoints) of all the training samples. (b) Second level of FSSC in $f_2$ feature-space (gaze direction) on cluster 2 and (c) cluster 3. Refer to Section 4.5 for the full details.

4.8 Confusion matrix for the 8 goals of the block placing experiment.

4.9 Confusion matrix for each individual block across the whole 160 experimental trials.

5.1 A summary of the experimental procedure by Vanderbilt et al. (2011) on ToM maturity in human children. A child would familiarize with an adult informant and subsequently ask for the position of a sticker hidden in one of two boxes, then decide whether to follow their suggestion or not.

5.2 The Bayesian network that models the relation between the robot and an informant. The agent generates a separate network for each user, with the same structure but different probability distributions.

5.3 Overview of the trust estimation architecture. The robot uses its sensors to identify each human and select their own BN or eventually generate a new one using episodic memory. The selected model is then used for inference.

5.4 Consistency values histogram of the episodes memorized by the agent through progressive interactions with different informants (from 100% helpful to 100% deceitful at 25% steps of variation. The opposite, that is from 100% tricky to 100% helpful, produces the same graph.)
5.5 Mean entropy of episodic memory networks generated with varying number of samples. The error bars represent the standard deviation of that batch. Given the random component intrinsic in the algorithm, a very large number of samples ($10^5$) have been generated for every value of $k$. 91

5.6 The experimental setup of the trust and ToM experiment. This printed mat marks the position of the robot and the informant, other than the two positions in which the sticker can be located and searched. 92

5.7 Familiarization phase with a tricker informant. (a) The robot asks for a suggestion on the sticker’s location. (b) The informant places the sticker in one of the two positions. (c) The informant gives its suggestion on where to find the sticker. Note how the tricker gives misleading directions. (d) The robot searches for the presence of the sticker in the suggested position and records the episode. 94

5.8 Decision making phase with a tricker informant. (a) The robot asks for a suggestion on the location of the sticker and receives a misleading suggestion from the informant. (b) The robot performs inference on that informant’s belief network. (c) The agent decides that the informant will probably try to trick it, so it looks in the opposite location. (d) The robot finds the sticker and gives feedback to the informant. 95

5.9 Belief estimation phase with a tricker informant. (a) The robot recognizes the informant using machine learning techniques and looks at the table to find the position of the sticker. (b) The robot computes inference on the informant’s belief network and predicts what he would suggest in that situation. 96

5.10 Reliability histogram of episodic belief networks generated by agents possessing different histories of interactions. Green bars represent trustful BNs ($T > 0$) and red bars depict BNs that tend to distrust ($T < 0$). Agents that have a more positive than negative background tend to be more prone to trust a new informant and vice versa. When $T = 0$ the informant is neither trusted nor distrusted and the agent will act randomly. 98
5.11 The proposed GWT model to predict trust in HRI. It is composed by two expert systems, both of which use different features to estimate the informant’s trustworthiness: the Experience Expert System (ExES) uses previous experiences of interactions, while the Engagement Expert System (EgES) makes use of social cues detected on the partner during the interaction. The final decision of the system will be moderated by the Global Workspace component.

6.1 Overview of the mechanisms leading to joint action. Expanded from Bauer et al. (2008) through the addition of Trust Estimation.

6.2 The collaborative workflow. The robot reads the intention and decides whether to trust its partner or not. In the former case, it provides assistance and evaluates the total output resulting from the collaboration, otherwise it adopts a more strict supervision on the human: if the partial output seems valid it gives them a chance to regain trust, otherwise it will take over the task and attempt to correct the mistakes.

6.3 A graphical representation of the simulated humans’ behavioral patterns. The deterministic agents are defined by the fixed sequence of interactions over time that they perform, whilst the stochastic group members are characterized by the success rate of their actions, which is then randomized at every iteration.

6.4 Comparison of the collaboration success rates with and without the trust model for each of the simulated informants.

6.5 Variation of the opinion value at each turn of interaction for the 3 deterministic simulated informants ($H_1$, $H_2$ and $H_3$), initialized with a trusting BN. When $O(H)$ becomes less than 0, the robot starts distrusting the informant and taking more control of the task.

6.6 Opinion dynamics for the stochastic human $H_4$ (50% success rate) during its first 10 iterations.

6.7 Opinion dynamics for the stochastic human $H_5$ (80% success rate) during 10 iterations.

6.8 Opinion dynamics for the stochastic human $H_6$ (20% success rate) during 10 iterations.

6.9 Opinion dynamics for the stochastic human $H_7$ (80% success rate, against a naively distrusting BN) during 10 iterations.
A.1 Skeleton decomposition for the goal “wall”. 140
A.2 Skeleton decomposition for the goal “tower”. 141
A.3 Skeleton decomposition for the goal “castle”. 142
A.4 Skeleton decomposition for the goal “stable”. 143

B.1 Training set for the intention reading experiment described in Chapter 3. 145
B.2 Training set for the “BGOR” goal from the intention reading experiment described in Chapter 4. 146
Abstract

Robots stand at the heart of a techno-scientific revolution which promises to alter the way in which we conceive our society. Recent discoveries point towards a future in which artificial agents will become fully integrated in our social structures, thus becoming important actors in our life. In this scenario, it will be critical for them to be able to understand us in the most human-like fashion and to assist us in our routines. We state that collaboration between humans and robots is fostered by two cognitive skills: intention reading and trust. The former is the capacity to discern the goal that is driving the actions of someone, while the latter is the ability to evaluate the trustworthiness of another agent. A robot endowed with these skills will be able to understand what kind and which degree of assistance its partner needs during a collaboration.

This thesis aims at advancing the scientific understanding of trust and intention compliant support in the interaction of humans and machines by presenting a robot learning architecture for collaborative intelligence based on the developmental robotics approach. We use probabilistic reasoning and a novel clustering algorithm that integrates multimodal social cues to infer the intention of the other agent, while estimating the partner’s trustworthiness through a Bayesian network and a novel episodic memory system. These two skills are then combined to formulate collaborative action plans.

We tested our models in human-robot interaction experiments involving joint manipulation tasks. The data we have collected demonstrate the effectiveness of our original methods, the importance of computational robotic models of human trustworthiness and, finally, the superior performance of collaborations involving trust estimations over ones based solely on goal prediction. Our results show that the synergistic implementation of these cognitive skills enables the robot to collaborate in a meaningful way, with the intention reading model allowing a correct goal prediction and with the trust component enhancing the likelihood of a positive outcome of the task.
Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.
Copyright

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the “Copyright”) and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the “Intellectual Property”) and any reproductions of copyright works in the thesis, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=24420), in any relevant Thesis restriction declarations deposited in the University Library, The University Library’s regulations (see http://www.library.manchester.ac.uk/about/regulations/) and in The University’s policy on presentation of Theses.
Acknowledgements

Whenever during the course of my life I have written some acknowledgements, I have always included my parents last. This is not to be intended as a lack of thankfulness but, on the contrary, it happened because my gratitude towards them is the deepest, more profound I can possibly conceive. Without thirty years of their constant, uninterrupted, unconditioned support I would have never embarked in the greatest adventure of my life. Without their love I would not be, today, half the person I am. Therefore, I decided that this time I would thank them for first, above anybody else.

I would then wish to thank Antonio Chella for believing in me when, for a moment, I stopped doing so myself and suggesting me to apply for this PhD program. Then of course, I would like to express my gratitude to my supervisors Angelo Cangelosi and Christian Goerick. The PhD process is a tough journey but you have managed to make it bearable thanks to your presence, your guidance and your teachings. I hope to live up to your high standards.

I would also like to thank the funding bodies that sustained my efforts: Honda Research Institute GmbH and the U.S. Air Force Office of Scientific Research, without which this work would not have been possible.

I thank my former colleagues both at the University of Plymouth and at the University of Manchester for sharing this path with me. You are many and I won’t name you all, but every one made a difference. I’ll cherish the time spent together and I’m grateful for some long-lasting friendships that I’ve made: having you spread across the globe makes the world a more friendly place.

Thanks to my life-long friends Domenico, Emanuele, Gioacchino, Giuseppe, and Marco for always being by my side even when tides have changed, scattering us in different places. Some bonds loosen with distance, but our one proved to strengthen even more. Whenever my life arrives at a crossroad, I can always count on you being my points of reference.

Last but not least I would like to express my love and gratitude to my partner,
Crefeda, for her life-changing affection, support, encouragement and company. Thank you for bringing happiness and adventure in my life. Thank you for not only being with me during the path, but for being the path and the destination themselves.
Chapter 1

Introduction

1.1 Motivations

Human beings are social creatures held together by communal bonds and organized into complex social structures, which provide fulfillment for many needs that range from basic survival to intellectual and emotional expression. This tendency to aggregation, to work as part of groups and to collaborate with others to achieve common goals is not to be dismissed as a quirk but rather it has been one of the key factors in the success of the human species. This characteristic has been proved being at least partially hardwired in our genes (Ebstein et al., 2010), explaining why even newborn babies possess the instinct and capability to perform social behaviors such as gaze following (Cangelosi et al., 2015).

From a cognitive perspective, the act of collaboration requires a shared understanding of common goals and intentions (Malle et al., 2001; Dominey and Warneken, 2011) and also mutual trust between the involved parties (Jones and George, 1998; Groom and Nass, 2007). Given this premise, we state that an intelligent agent able to perform collaborative behaviors must be endowed with at least two fundamental cognitive skills: intention reading and trust estimation. The former is the ability that bootstraps social awareness and it consists of using observable social cues such as eye-gaze, gestures, motion and speech to interpret other agents’ actions and assign them meaning. The latter is the ability to evaluate how trustworthy or reliable another agent is expected to be with respect to a certain task, particularly useful during joint action scenarios where a group of agents depend on each other’s efforts to achieve a shared goal. The synergistic use of these two cognitive functions would enable a collaborative agent to determine what kind and which degree of help should be offered to a partner in order
1.1. MOTIVATIONS

to optimize its chances of success.

Given the importance of collaborative behavior for humans, it seems natural to transfer its value to artificial agents, in particular to social robots which are expected to act in human-shaped environments, interacting with us on a daily basis. In particular, if we aim at designing robots capable of blending themselves optimally into our present and future societies, a strict requirement for them will be to adapt to our social expectations and fit in our natural environments, which will be possible only if they possess our same social skills: this is considered the key to create truly social machines (Breazeal, 2004). For example, if a humanoid robot and a person are engaged in joint action, both must be able to understand the goal and intentions of the partner, dynamically adjust their behavior to the shared goal and communicate the intended objective to the other agent. This dynamic, two-way cooperation is one key aspect of collaborative intelligence between people and machines, requiring embodied and situated interaction between humans and robotic agents (Sciutti et al., 2015). In other words, in a future where interactions between humans and robots will be more common, we don’t want to robotize people, but we aim to make the minds of these mechanical companions increasingly more human. For this purpose, collaborative intelligence may be one of the most important skills for these agents to possess.

This thesis presents a series of works based on the developmental robotics paradigm, defined as “the approach to the design of behavioral and cognitive capabilities in artificial agents that takes direct inspiration from the developmental principles and mechanisms observed in the natural cognitive systems of children” (Cangelosi et al., 2015). In other words, our computational models are based on scientific findings in human cognition and take inspiration by the psychological development of human children. Our main contribution comes in the form of a novel cognitive artificial architecture for human-robot collaboration capable of performing both intention reading and trust estimations on human partners. To achieve this, we have made use of a set of state-of-the-art techniques ranging from probabilistic models to unsupervised machine learning methods, including a novel multimodal clustering algorithm. We have validated this architecture and its individual components through a set of both physical and simulated human-robot interaction (HRI) experiments involving a series of cooperative tabletop object manipulation games. Our results not only prove the effectiveness of our new methodologies to infer intentions and trustworthiness from humans, but also support the idea that trust is a key element to achieve high performances during team collaborations by providing the robot with some decision-making parameters that are used to
fine-tune the assistive behavior.

1.2 Research Questions

Having framed the scope and motivation that drive this line of research, here we formulate the main questions that we wish to address through this thesis.

RQ1. Can a robot learn to infer intentions from a human partner through the observation of behavioral and social cues, using lightweight and unsupervised methodologies which do not involve large datasets, long training times or hand-crafted goal libraries?

RQ2. Trust is a critical issue in HRI, but it has mainly been studied from the human’s perspective. What would happen in the reverse scenario, where a robot is the trustor and a human is the trustee? Can a robot learn to dynamically estimate the trustworthiness of a human partner and act based on that prediction?

RQ3. Based on current psychological theories, we state that collaborative intelligence arises from the mutual interaction of two cognitive skills, namely intention reading and trust estimation. Is this true? Can a robot use these faculties to assist a human partner? Does a collaboration involving trust perform better than one based solely on goal prediction?

1.3 Aims and Objectives

To address RQ1 and RQ2, this thesis proposes a set of specialized artificial cognitive architectures designed to enable intention reading and trust estimation on a range of social robots. The research methodology seeks to:

1. Survey the current state-of-the-art in the psychological domain to identify the cognitive, developmental and neuroscientific origin of the biological mental skills under examination. This will allow us to understand how these abilities arise in natural intelligence and give us insights on how to develop them in a robotic agent.

2. Conduct a literature review of the computational models that already exist in the scientific and engineering domain. This will lead to the identification of a number of models, their potentials and the limitations we aim to overcome.
3. Design and implement a cognitive model for an embodied robotic agent.

4. Empirically evaluate our model through physical HRI experiments.

To answer RQ3, the two architectures resulting from the investigation of the previous research questions will be integrated in a unified computational model able to perform collaborative intelligence. In particular, our methodology aims to:

1. Survey both the psychological theories behind biological cognitive collaborative intelligence and the computational models in existence for virtual and embodied agents.

2. Design and implement an integration of our intention reading and trust models in a single architecture able to synergize these two cognitive abilities to perform collaborative assistance.

3. Perform simulated experiments and compare different models to investigate the impact of trust estimation in support of intention reading during collaborative HRI.

1.4 Contributions to Knowledge

This section summarizes the main contributions of the thesis with reference to the related publications and the chapters in which they are discussed. We will revise the following in Chapter 7, after having presented the details of our research.

1. We initially explore intention reading in HRI, designing a cognitive architecture which is able to overcome some of the limitations of the state-of-the-art computational models, such as the rigid dependency on large datasets or handcrafted goal libraries. Following some psychological theories, we propose a double-layered architecture which executes low-level action recognition through a dynamical clustering of the human’s body postures, paired with a high-level goal prediction executed through a Hidden semi-Markov model. Publication 2 in Section 1.5 is related to this contribution, which is described in detail in Chapter 3.

2. To address some of the limitations of our preliminary intention reading model, we have expanded it to take into account more social cues other than body posture. To do so, we propose Feature-Space Split Clustering, a new clustering
algorithm that has been developed to differentiate multi-sensory human social cues by performing several levels of clustering on different feature-spaces. This more advanced architecture is discussed in Chapter 4 and in publication 4 in Section 1.5.

3. This thesis contributes to the literature of trust in HRI by providing the design and implementation of a robotic architecture able to estimate trust in humans involved in joint action. We based our study on a psychological experiment by Vanderbilt et al. (2011), which aimed at evaluating the Theory of Mind (ToM) maturity in children of different ages, that is their understanding of other agent’s mental life. Our computational model, consisting of a Bayesian network that integrates both trust and ToM, was designed with the aim of being able to reproduce the results of the original experiment. The robotic architecture also introduces a novel episodic memory algorithm based on information theory and weighted sampling. Publications 1 and 3 in Section 1.5 refer to this contribution, which is also detailed in Chapter 5.

4. Our last contribution comes in the form of a cognitive architecture for collaborative intelligence in embodied robotic agents, which joins our intention reading and trust models and directs their efforts in providing assistance in a shared activity with a human partner. Through the use of this computational model we were able to demonstrate the positive influence of trust on the synergistic efforts of the two agents. Publication 5 in Section 1.5 and Chapter 6 focus on this specific contribution.

1.5 Publications

A list of publications, either published or in the process of review, that relate to the contributions of the thesis is given below:

1.6. THESIS STRUCTURE

This thesis is composed of the following 7 chapters:

Chapter 1 provides an introduction to the problem under examination and a motivation for this research, other than framing the major questions that we aim at answering. It also states the expected contributions made by the author and lists some selected publications that have been produced.

Chapter 2 provides a literature review on HRI with a focus on collaborative intelligence. This chapter discusses intention reading and trust estimation both from a psychological and a computational point of view: this will allow the reader to be aware


- **Publication 4**: Vinanzi, S., Cangelosi, A., & Goerick C. (2020). “The Role of Social Cues in Human-Robot Cooperation”. 29th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN) (pp. 971-977). (Vinanzi et al., 2020). This paper builds on our previous model of intention reading and demonstrates the use of our novel Feature-Space Split Clustering algorithm for multimodal intention reading.


1.6 Thesis Structure

This thesis is composed of the following 7 chapters:

Chapter 1 provides an introduction to the problem under examination and a motivation for this research, other than framing the major questions that we aim at answering. It also states the expected contributions made by the author and lists some selected publications that have been produced.

Chapter 2 provides a literature review on HRI with a focus on collaborative intelligence. This chapter discusses intention reading and trust estimation both from a psychological and a computational point of view: this will allow the reader to be aware
of the biological cognitive foundations on which we ground our methodology and the current state-of-the-art to which we aim to contribute. Other than that, we link our methods to specific limitations in the current literature.

**Chapter 3** introduces our first intention reading HRI model. This cognitive architecture is based on the observation of the partner’s physical movements and performs a low-level action representation paired with a high-level goal recognition. Here we describe our first HRI experiment involving an iCub robot.

**Chapter 4** builds on the foundations of the previously introduced intention reading model and expands it by adding support for multimodal perceptual inputs by using a novel algorithm, Feature-Space Split Clustering. We describe a new experiment, involving a Sawyer robot, where we scale up the complexity of the setting and use both the partner’s posture and gaze to infer their goals.

**Chapter 5** describes a unified trust and ToM model and the psychological experiment from which it was derived. In this work we embedded a Pepper robot with a Bayesian network that simulates the mental abilities of a 5-year-old human child to make it capable to distinguish reliable from unreliable informants.

**Chapter 6** describes the final experiment for this research project in which the intention reading and trust models are integrated in a computational architecture for collaborative intelligence. This step aims to demonstrate the positive influence of trust on the interaction performance.

**Chapter 7** concludes the thesis with a summary of what has been done, the scientific and technological insights that have been gained through this line of research, some of its limitations and, finally, possible future directions that we hope to inspire in the scientific community.
Chapter 2

Background

2.1 Introduction

This thesis is based on work done in the field of cognitive robotics, which aims at creating intelligent behavior in a robot by providing it with a computational architecture able to learn and reason about how to behave in a complex world. Vernon et al. (2011) state that “cognition is the process whereby an autonomous self-governing system acts effectively in the world in which it is embedded” and that its primary role is to anticipate future events in order to compensate for the latencies inherent in the processing of both the senses and the physical actions.

This chapter introduces the theoretical background of this research and the relevant literature domain. However, for the sake of clarity, some topics that are related to specific experiments are presented in the methodological section of their respective chapter. The following sections are organized as follows:

- Section 2.2 presents a review on collaborative intelligence, both from the psychological and computational points of view.

- Section 2.3 focuses on intention reading, first covering its cognitive and neurological origins, then shifting to some relevant artificial intelligence models in the current literature.

- In Section 2.4 we present trust and Theory of Mind (ToM) in HRI, with a focus on robot-side human trustworthiness estimations.
• Section 2.5 provides an overview on the field of developmental robotics and introduces the robotic platforms that are used within this research.

• Finally, Section 2.6 provides a summary of the limitations of the state-of-the-art models and presents our plan to overcome them.

2.2 Collaborative Intelligence

2.2.1 Cognitive Foundations

The importance of society within our species is undeniable, as it provides fulfillment for many of our needs: from basic survival to mental and intellectual development, including the expression of a wide range of emotions. The reason why humans started gathering in groups is mainly evolutionary, as isolated individuals would be more endangered against the wild, but what really made it possible is the intrinsic attitude towards social abilities we possess as a species. It could be said that our evolution and prosperity are rooted in our collaborative intelligence, which is defined by Gill (2012) as the cognitive ability that makes an agent (human or machine) able to autonomously contribute to a problem-solving network.

Researchers in the social sciences agree to distinguish collaboration from cooperation, as they represent two different types of interaction (Roschelle and Teasley, 1995). In particular, we refer to “cooperation” when the involved parties work towards a shared goal by solving sub-tasks individually and then assembling their partial results. In contrast, “collaboration” refers to the act of dividing the task among the participants, who then engage in a mutual, coordinated effort to solve the problem together. Given these definitions, the main difference between cooperation and collaboration is that the latter implies a deeper level of interaction, shared understanding and coordination (Dillenbourg, 1999).

A body of scientific evidence points towards the early development of collaborative behaviors in human infants: the latter are, in fact, able to engage in coordinated actions as early as their first birthday. This ability continues to evolve through time and by experience, in parallel to their cognitive development, and by the 30th month of age they become able to perform complementary actions (Henderson and Woodward, 2011).

Our hypothesis on collaborative intelligence stems from two statements. Bauer et al. (2008) break the collaboration process into a series of sequential tasks, namely:
perception, intention estimation, planning and joint action. In other words, before an
agent can collaborate with another there is the need of recognizing the pursued goal and
to select appropriate actions to maximize the chances of a successful outcome. Groom
and Nass (2007) declare that trust estimation is an essential component to successfully
perform joint activities with common tasks. From these premises, we state that the two
cognitive skills essential for successful collaboration are intention reading and trust.

2.2.2 Computational Models

The term “collaborative intelligence” was coined in the context of computer science
to denote heterogeneous multi-agent problem-solving systems. One of the precursors
of this kind of technology was the Pandemonium Architecture (Selfridge, 1959) which
used a blackboard system to mimic several neurological areas of the human brain with
the purpose of letter identification.

In the context of robotics, collaborative intelligence models and applications fall
under the umbrella term of human-robot collaboration (HRC) which is a sub-field of
the more general HRI. Effective HRC requires the robot to possess several communica-
tion mechanisms in order to both understand humans and to inform them about
its own goals, and in so doing maintain a set of shared beliefs which support the exec-
tion of a joint plan (Hoffman and Breazeal, 2004). The state-of-the-art of cognitive
robotics presents numerous models of HRC (Bauer et al., 2008) which demonstrate the
use of a wide range of techniques for different kinds of settings. A successful example
of one of them is provided by Pineau et al. (2003), who employed a mobile robot for
elderly care that possesses a dialogue manager to communicate with its patients and
that is also able to perform probabilistic planning and reasoning. Another instance of
collaborative robot is Robonaut: a humanoid robot developed by NASA (Bluethmann
et al., 2003) which is designed to assist astronauts on the International Space Station.
Its cognitive architecture heavily relies on sensing, and it is able to understand human
speech and to perform manipulation tasks, such as working with tools during both
intra- and extra-vehicular activities to support the human crew. Lallée et al. (2010)
have designed a framework that allows verbal cooperation with a set of robots based
on imitation and experience: for example, an iCub robot was able to assist its partner in
the assembly of a table, holding one part in position whilst the human attached another
component to it. In this case, the robot did not learn the primitive actions, which were
already known, but rather how to use them in order to achieve a goal. Other fields of
application involve medical assistance during rescue operations (Murphy et al., 2004)
CHAPTER 2. BACKGROUND

and creative artistic performance (Giardina et al., 2017).

All these examples comply with the previously mentioned requirement of communication mechanisms with the human partners, but another important factor is safety: if these machines are designed to operate side-by-side with humans, it is critical that they avoid as much as possible to endanger them (Cherubini et al., 2016). More generally, the architectures of current collaborative robots seem to mostly follow the design formalized by Bauer et al. (2008) which focuses heavily on sensing: the robot observes the human and tries to understand their intention, which can be expressed explicitly through speech, gesture or haptic communication or implicitly by the means of social cues such as actions, expressions and gaze direction. Once an end objective is predicted, the robot has to perform some planning to generate a sequence of appropriate collaborative actions that are finally executed. The robot is then required to not only read intentions, but also to share them: collaboration is not a one-way interaction but rather a bidirectional effort from both the involved parties. The robot has to communicate its own beliefs about the common goal and the plan in order for the human to be aware and eventually adjust or correct them.

An interesting experiment by Dominey and Warneken (2011) involved the use of a robotic arm to play a turn-based game where the human and the robot jointly construct a shared plan that allows them to move some puzzle pieces around a table. The artificial agent is equipped with vision capabilities and the ability to understand natural language to extract predicates which are in turn translated into primitive actions. Through a series of experiments, the authors of this study have demonstrated how their cognitive architecture allowed the robot to imitate actions and to construct shared plans where the tasks are alternatively delegated to itself or the human. In addition, the robot is able to manage cases of role reversal or situations in which the partner suddenly stops collaborating.

These few examples taken from the wider literature on HRC demonstrate that there is no strict and formal definition of how a collaborative architecture should be defined. The robot might inherently possess a model of the world and of its actions in the form of motor primitives or it could learn them from the observation of the environment and its actors. Moreover, it might expect explicit communication or be required to autonomously understand an already ongoing action. As we will discuss in the next section, even the mathematical and computational models used to achieve the sub-objectives required for the collaborative effort can vary widely based on the context and setting in which the robot is required to operate.
Having stated in Section 2.2.1 our hypothesis on collaborative intelligence as an emergent behavior which stems from the synergistic display of intention reading and trust estimation capabilities, we will now introduce these two cognitive mechanisms in detail, both from a developmental, biological perspective than from a computational one.

2.3 Intention Reading

2.3.1 Cognitive Foundations

Both adults and children possess the innate capability of understanding the behavior of others not as a series of unrelated motions through space, but rather as sets of goal-directed actions (Malle et al., 2001). In fact, humans are able to see observed actions as a continuous stream of movement that is divided in intervals which are then segmented and interpreted (Baldwin and Baird, 2001). The cognitive ability that makes us decode the purpose of someone’s actions is called intention reading: a mental skill that lays the foundations of our social awareness (Woodward et al., 2009) and enables us to reason about other agents in our environment to perform appropriate decision making. Once perceived, intentions can be shared and used to coordinate joint actions with common goals (Tomasello et al., 2005). Overall, this cognitive ability enables us to be social creatures.

There are two ways in which an observed action can be interpreted: it can be teleological if it is understood as goal-oriented, or referential if its aim is to ascribe attentional states or communicative messages (for example, pointing gestures) (Csibra, 2003). More generally, intentional action can be viewed as a closed loop process in which a goal is formulated (for example, to open a bottle) and, by integrating perception of the environment, knowledge and skills, an intention is generated (grasp the bottle and remove the cap). Subsequently, the plan will be performed and the effects of the actions will be perceived and used as feedback (if the cap is hard to remove by hand, retrieve a tool). The previous example also demonstrates the hierarchical organization of intentions: some primitive actions may complete goals that serve the purpose of a higher goal (to drink the content of the bottle) (Baldwin and Baird, 2001).

A neuroscientific explanation of the reasons why humans are so good at mind-reading is the mirror neuron system (Rizzolatti and Sinigaglia, 2007; Gallese and Goldman, 1998). These are neurons originally discovered in the F5 cortical area of
the macaque brain that activate both when performing a specific action and when observing others perform it, as in an unconscious preparation for imitation. By providing the physiological mechanism for the perception/action coupling, this system is important for learning new skills by imitation. This happens because actions are intrinsically goal-oriented, meaning that the brain possesses a mapping between goals and actions, the latter encoded as a set of neural activations. By experiencing the same neurological motor responses of the observed agent, the brain is able to reverse-engineer their intended goal. Compared to the animal mirror system, the human one appears to have developed some morphological differences driven by evolution: it is, in fact, able not only to understand the goal of observed motor acts but also to derive the intention behind them (Fabbri-Destro and Rizzolatti, 2008). For example, a study by Iacoboni et al. (2005) proved that mirror neurons could discern whether another person who was picking up a cup of tea planned to drink from it or clear it from the table. It has been theorized that the mirror neurons develop gradually with time through a Hebbian learning process, meaning that the strength of the connections between these neurons are modulated during development through experience (Kilner and Blakemore, 2007).

The Hebbian development theory is backed by further evidence collected on a behavioral rather than neurological level, since many researchers have proved that preschoolers, whose intention reading ability is typically tested using the visual habituation paradigm (Malle et al., 2001), learn to derive goals from actions via a direct and progressively refined experience of the world (Woodward et al., 2009). In particular, social cues such as biological motion and eye gaze (Tomasello et al., 2005) are used to achieve an increasingly refined understanding that evolves across various stages of development: what is understood by children is originally biological motion, then the pursuit of goals and finally the choice of plans. By the eighteenth month of age, evidence suggests that children are skilled enough to understand intentions behind failed actions and re-enact a successful version of them (Baldwin and Baird, 2001; Meltzoff, 1995). A milestone in development seems to be the fifth year of age, when intention reading reaches its maturity (Tomasello et al., 2005). Not surprisingly, this is the same age at which children fully develop ToM: the ability to understand that others possess different mental states (such as knowledge and goals) from one’s own (Vanderbilt et al., 2011). This makes sense, as the two mental skills overlap: intentions cannot be read if the beliefs and desires that drive the other are not acknowledged. ToM will be discussed more extensively in Section 2.4.
2.3. **INTENTION READING**

2.3.2 **Computational Models**

The scientific community has been researching intention reading capabilities in robots for many years. Two main categories of research have emerged: one concerning the replication of the mirror neuron system and one that takes a more cognitive and less neuroscientific approach. The former is composed of a set of techniques that aim at replicating low-level imitation capabilities in an artificial agent rather than investigating intention reading itself. Several computational models have been developed in the last decade for this purpose: recurrent neural networks, genetic algorithms, dynamic systems and neuromorphic architectures (involving spiking neural networks) are just a few examples of methodologies that have been adopted (Oztop et al., 2006).

In the scope of this thesis, we adhere to the higher-level cognitive approach which has itself a rich literature. Dominey and Warneken (2011) explored shared intentionality in a turn-taking game between a human and a robotic arm, where the artificial agent would build a representation of the shared plan and subdivide the actions between itself and its partner. Bien et al. (2005) developed a system that analyzes posture and movements in elder people and tries to decode the inner intentions they are driven by. Sciutti et al. (2015) affirm that humans are sensible to subtle kinematic differences that can distinguish similar actions directed to different goals and proposes a robotics approach to investigate this. Other relevant researches involve the use of variations of dynamic Bayesian networks (Tahboub, 2006), self-organizing maps (Buonamente et al., 2015) and first order logic (Jansen and Belpaeme, 2006). In an experiment by Duarte et al. (2018), several social cues such as saccadic eye movement, gaze directing and arm movements are tested towards the action anticipation capabilities of a humanoid robot. Here the humans are equipped with a suite of sensors and a Gaussian Mixture Model is used to model the trajectories of their movements from the robot’s perspective. Another widespread methodology consist in using Partially Observable Hidden Markov Models, which are often used to track and anticipate the movements of a human across different rooms of a building, implicitly inferring his or hers intentions based on their destinations (Ognibene et al., 2019). Other less common approaches include the use of Latent Semantic Analysis to analyze the relationships between spatio-temporal features extracted from video clips (Duckworth et al., 2016).

Granada et al. (1995) propose an approach where the intention recognition task is divided in two levels: the low-level is focused on action recognition and classification, whilst the high-level derives the goal from the observed actions. Action recognition is a topic that has received great attention from computer vision researchers (Krüger et al.,
CHAPTER 2. BACKGROUND

and which is deeply linked to intention reading. The most consolidated state of the art techniques involve the use of a modern deep learning architecture consisting of a convolutional neural network (CNN) connected to a recurrent neural network, often a Long-Short Term Memory (Donahue et al., 2015) that takes as input an image frame from a video sequence and outputs a label representing the recognized action (Singh et al., 2016; Baccouche et al., 2011; Shi et al., 2016). Other similar approaches work not by operating on an RGB image but on skeletal data extracted by means of computer vision algorithms as OpenPose (Cao et al., 2016) or by depth sensors such as the Microsoft Kinect (Manzi et al., 2017). Plan recognition, in contrast, is usually performed by means of probabilistic modeling such as static or dynamic Bayesian networks (Tahboub, 2006), first order logic (Kautz and Allen, 1986) or hybrid architectures (Dindo and Chella, 2013). These models often assume the availability of a plan library capturing the system’s knowledge about goals and means of accomplishing them (Carberry, 2001).

2.4 Trust and Theory of Mind

2.4.1 Cognitive Foundations

Trust is a fundamental, unavoidable component of social interactions that can be defined as the willingness of a party (the trustor) to rely on the actions of another party (the trustee) with the former having no control over the latter (Mayer et al., 1995). In other words, it represents the willingness of the trustor to handle over the responsibility of a task to the trustee and to put the outcome at risk. Trust is involved in every sort of social interaction and is a key factor in the achievement of successful relationships, in our personal safety (Das and Teng, 2004) and in team cooperation (Jones and George, 1998): misplaced trust can, in fact, result in economical, emotional or physical damage. For example, when someone boards an airplane they inherently decide to trust the pilot to perform its job safely and correctly. Again, a bank approving a loan is trusting the economical ability of the loanee to return the investment. If either the airplane passenger or the bank have misplaced their trust in their trustee, they will have to face the consequences of their choices.

The concept of trust is directly linked to the one of reliability, that is the predictability of an agent. Being predictable is not inherently a positive virtue: an agent can, in fact, be reliable in its failures. Therefore, trust is not mere reliance, rather the belief
that the trustee is both positively reliable and committed towards the task (Hawley, 2014). At the same time, distrust is not the mere absence of trust but, instead, the belief that the other party is committed and at the same time non-reliable.

Regarding the biological origin of trust behavior, there is some evidence that it is at least moderately genetically predetermined and that some hormones, in particular oxytocin, and some brain structures are strongly connected with one’s personal trust dynamics (Riedl and Javor, 2012). In contrast, its development during childhood is still under debate by developmental psychologists. Erikson (1993) theorized that infants not older than two years pass through a stage known as “trust vs mistrust”, where their propensity to trust is shaped by the quality of care received. This happens because infants highly depend upon the caregivers for sustenance and learn whether or not the latter regularly satisfy their basic needs, either learning that the world is a secure, trustable environment or an undependable, insecure place. Evans et al. (2013) has found empirical proof that the ability to trust is linked to age, as older children seem able to take another person’s perspective to gain insights for behavioral prediction.

A psychological trait that relates to the mastery of one’s self-trustfulness is ToM: the ability to attribute mental states to others (for instance, beliefs, intentions and desires), that may differ from one’s own. In fact, the ability to correctly judge the trust-worthiness of others is strongly correlated to the maturity of this trait (Rotenberg et al., 2015) because a mature ToM allows to compare the other agent’s beliefs and motivations with one’s own to verify their alignment (Premack and Woodruff, 1978). Despite ToM being universal in adults, the same cannot be said about preschoolers: whilst the latter are not completely lacking some form of ToM, this slowly develops with age (Koenig and Harris, 2005; Wellman et al., 2001). A famous and historical test that aims at measuring the ability of someone to attribute false beliefs to others is the Sally-Anne test (Baron-Cohen et al., 1985): a subject is introduced to two characters, Sally and Anne. The former puts a marble in a basket and then leaves the room. While she’s missing, Anne takes the marble from the basket and puts it into a box. At this point Sally is reintroduced in the scene and the subject is asked where will she look for the marble. A developed ToM will let the subject simulate the mental states of the characters and understand that Sally has her own (false) beliefs which do not necessarily align with one’s own or the reality: in this case, the test is passed if the subject answers that Sally will look for the marble in the basket, where she believes it still is. On contrary, if the subject has an immature ToM they will believe that their own mental states and perspective are shared by everyone else, hence the wrong assumption that Sally will
search for her marble in the box.

In summary, ToM is a cognitive skill that undergoes a developmental process and needs time to fully mature. An experiment conducted by Vanderbilt et al. (2011) on the ability of children to attribute trust or distrust in others has demonstrated that ToM matures around the fourth year of age and is completely developed by the fifth year.

2.4.2 Computational Models

Whereas trust is such an important factor in human interactions, it is also an essential component of HRI, in the sense that a great degree of trust improves the quality of interactions with the robot and, vice versa, successful interactions enhance the machine’s trustworthiness from the user’s point of view. This means that even a perfect machine won’t be able to perform at its fullest if the human partner is not willing to trust its decisions and actions (Knott et al., 2018). This phenomenon is named “social influence”: it is easier to influence or persuade someone who is trusting. In human and robot teaming scenarios, where the two share a common goal, trust is an essential component to successfully perform joint activities (Groom and Nass, 2007).

Hawley (2014) states that inanimate objects can be reliable but not genuinely trustworthy. This means, for example, that I can rely on a shelf to be able to withstand a certain weight, but I would not feel betrayed by it in case it broke. This statement does not apply to humanoid robots, which are a very special case of objects: in fact, despite their artificial nature they can communicate and interact and, by adopting the participant stance, can be worthy of trust or distrust (or neither) from their human partners (Holton, 1994). This has important implications in human-robot teaming as the machine has to comply with a set of functional, agency-based, appearance-based, social-relational, and existential criteria that wouldn’t be necessary for the adoption of a tool or instrument (Coeckelbergh, 2012).

This problem has generated a branch of research focused on determining which behavioral and aesthetic elements of a robot can influence its perception from the people who interact with it (Hancock et al., 2011), in other words there is a vast literature on human-centered trust in HRI (Floyd et al., 2014; Zanatto, 2019). Here we propose that the opposite, i.e. the trustworthiness of a human estimated by a robot, is also fundamental during a collaborative activity: whereas a robot can fail, so can a person and it is important to keep this in mind when performing decisions that will try to optimize the achievement of the shared goal. Unfortunately, to the best of our knowledge literature is scarce for what concerns this kind of robot-centered trust. There are a number of
numerical models that use time-series analysis and dynamic Bayesian networks to determine a quantitative trust factor for a human interacting with a robot (Sadrfaridpour et al., 2016; Xu and Dudek, 2015), but usually this value is used to calibrate the robot’s behavior for a better acceptance by its users. Rahman et al. (2015) propose a model of bilateral trust dynamics in which a robot and a human involved in a joint task estimate each other’s trustworthiness based on their performance on the activity by means of a dynamic system. On the other hand, ToM models are much more common and they are frequently based on probabilistic modeling (Hiatt et al., 2011) or first-order logic (Devin and Alami, 2016) but they are mainly used to infer the human’s intentions rather than his or her trustworthiness.

Patacchiola and Cangelosi (2016, 2020) have presented a probabilistic model which unifies trust and ToM to be used in a simulation of Vanderbilt’s experiment about children’s trust willingness (Vanderbilt et al., 2011). In this thesis, we expand upon this computational model embedding it in a cognitive architecture for humanoid robots enhanced with an episodic memory system. The latter is a subcategory of the long-term declarative memory that stores memories about temporally dated episodes or events and temporal-spatial relations among them (Tulving et al., 1972). This feature is relevant because the positive influence of one’s personal history on the cognitive capabilities has been proven other than for the biological brain also for artificial agents (De Castro and Gudwin, 2010; Jockel et al., 2008). Episodic memory is also the key to reproduce the “trust vs mistrust” stage theorized by Erikson (1993) in a developmental cognitive system, which in this research is used in experiments about character formation for the robot.

2.5 Developmental Robotics

2.5.1 Definition

Developmental robotics (also known as epigenetic or ontogenetic robotics) is a scientific discipline which stemmed from the idea that an artificial agent might build better models of itself and the world if it developed them autonomously instead of having them imposed by a designer. A formal definition has been given by Cangelosi et al. (2015): “developmental robotics is the interdisciplinary approach to the design of behavioral and cognitive capabilities in artificial agents that takes direct inspiration from
the developmental principles and mechanisms observed in the natural cognitive systems of children”. In other words, the approach taken by this paradigm is to provide robots with the necessary tools for the acquisition of high-level cognitive skills by mimicking the same developmental processes we observe in humans. For example, whereas a classical approach to robotics would endow a humanoid robot with a mathematical model of gait to allow it to walk, a developmental system would rather create one autonomously using motor babbling. This approach has some commonalities with evolutionary robotics, but whereas the latter focuses on the phylogenetic development of a population, developmental robotics capitalizes on the ontogenetic evolution of one individual.

This branch of science has a strong interdisciplinary nature and it is made possible by connections between social sciences, such as psychology, neuroscience, linguistics, and computational disciplines such as robotics and artificial intelligence. This interconnected approach allows the validation of cognitive theories on robots to achieve a better understanding of the human being and, through this acquired knowledge, the design of better robots in a virtuous circle which benefits both the social and the computational sciences.

In the pursuit of the developmental robotics approach, the intention reading model presented in this thesis lacks a pre-existing plan library, rather it follows the psychological theories which state that this cognitive ability is learned by experience (Malle et al., 2001). Furthermore, it follows the principles theorized by Tomasello et al. (2005) which state that the intention decoding task is divided into a low-level action understanding based on social cues and a high-level goal prediction. The trust model also complies with this paradigm and was developed to recreate the attitudes of a child subject to the ToM maturity experiment by Vanderbilt et al. (2011).

### 2.5.2 Robots for Developmental and Cognitive Research

Vernon et al. (2011) state that: “cognitive systems are intrinsically embodied and embedded in their environment in a situated historical developmental context. Furthermore, the system’s physical embodiment plays a direct constitutive role in the cognitive process”. This means that the physical shape of a robot directly impacts its ability to interact with the world and so both the mental and physical skills it will develop. Given this premise, if the aim is to design a robot that embodies a human-like cognition, it is necessary to provide it with a human-like body. This is why in this thesis we have only used humanoid or semi-humanoid platforms, in particular those shown
2.5. DEVELOPMENTAL ROBOTICS

![iCub, Sawyer, Pepper](image)

**Figure 2.1**: The humanoid robots used in this research. Sources: Italian Institute of Technology, Rethink Robotics, SoftBank Robotics.

in Figure 2.1 and presented below.

### 2.5.2.1 iCub

iCub (Figure 2.1a) is an open-source humanoid research platform that resembles a three and half year-old child which was developed by the Italian Institute of Technology to study embodied cognition (Metta et al., 2008). With its 53 degrees of freedom (DoF), this robot possesses enough dexterity to perform social tasks such as gaze following, object manipulation, gesturing, crawling and walking, while being able to perceive the world through its stereoscopic cameras and its tactile skin. Its software library uses YARP, a middleware that connects its sensors, actuators and processors, for internal and external control.

### 2.5.2.2 Sawyer

Sawyer (Figure 2.1b) is an industrial collaborative robot designed by Rethink Robotics for object manipulation and manufacturing tasks alongside humans. It is equipped with a 7 DoF arm which can mount different kind of grippers based on the task it is employed for. This robot can sense the world through its multiple RGB cameras and force sensors. The software architecture is deeply interconnected with the ROS middleware.
2.5.2.3 Pepper

Pepper (Figure 2.1c), by Softbank Robotics, is a humanoid robot designed for social interaction in human environments. It is not a functional robot, meaning that it lacks the dexterity to perform manipulations, rather it is intended to have a more communicative role by optimizing its interactive capabilities. This is facilitated by its wide range of sensors, including cameras, microphones and touch sensors. It possesses 20 DoF and it is powered by the NAOqi operating system.

2.6 Research Plan

Having presented the current state-of-the-art, we identify some of the most common limitations which we wish to overcome:

- Many intention reading models depend on either a hand-crafted goal library or on supervised machine learning. The problem with these methods is that they lack scalability, meaning that if the agent has to learn about a new goal, either the plan library must be updated by the designer (who may very well not be the end user) or the system needs to obtain new, possibly numerous, samples and undergo a long training process.

- The current literature makes an extensive use of a range of sensors which are either equipped on the human to collect their physical features or embedded in the robot. Some of the latter, for example RGBD cameras, are not always available on every kind of platform and their lack is often compensated by adding hardware to the experimental setups (for example, placing one or multiple Microsoft Kinect to collect three-dimensional image data). We want to avoid the use of specialized sensors and work with the kind of hardware that is included in the majority of robots: RGB cameras. Additionally, we want to avoid placing sensors on the humans as this is not, in the author’s opinion, the most natural way to perform real-life HRI.

- The problem of quantification of human trustworthiness from the point of view of the robot is of critical importance in the scope of HRC: a machine able to predict the failures of its partner will have a better chance at ensuring the successful achievement of a shared goal by adopting appropriate corrections. Nevertheless
there are not many models in the current literature that consider this aspect and trust is a topic which is primarily considered from the human’s perspective.

- Due to the lack of models of robot trust, the state-of-the-art is, to the best of our knowledge, not considering collaborative cognitive models which take this factor into account, despite it being acknowledged as one of the requirements for successful teaming (Jones and George, 1998).

Given these open problems, we propose a set of models for intention reading, trust estimation and collaborative intelligence that try to overcome these limitations by using a developmental approach. These cognitive architectures are presented in detail in the next chapters.
Chapter 3

Intention Reading

3.1 Introduction

This chapter introduces the reader to our first intention reading model, which predicts goals based on the observation of human movements using unsupervised machine learning and probabilistic Markovian models. This architecture is used in a joint action context where a human and a robot need to cooperate to reach a shared goal. In such a scenario, the role of the robot will be to observe its human companion initiate a task, infer the underlying intention and offer to collaborate, intervening in the ongoing plan. In the pursuit of the developmental robotics approach (see Section 2.5), the system was not provided with a pre-existing plan library, conversely it was designed following the psychological theories which state that intention reading is developed through experience in the very first months of life (Malle et al., 2001). It also follows the low- and high-level subdivision of actions and goals theorized by Tomasello et al. (2005), performing a low-level body movement analysis that provides inputs for a high-level goal determination mechanism. Our approach is supported by evidence that infants recognize biological motion and use it as the main clue to read others’ intentions (Bertenthal, 1996).

This experiment represents our first attempt to create an artificial intention reading cognitive model, which is furthermore expanded in Chapter 4. Our main contribution comes in the form of a novel architecture that combines developmental theories with lightweight methods which allow the robot to generate a knowledge base on-the-fly, with no need of preexisting goal libraries, big datasets or long training times.
3.2 Methods

3.2.1 Clustering

Clustering is a data analysis technique that consists in dividing a set of data points into a number of groups, named clusters, such that each cluster contains data points more similar to each other than the ones belonging to other groups. The main application of this methodology is pattern recognition, which is the automated identification of regularities in data, widespread across a vast and diverse array of applications. For example, clustering is adopted in banking (fraud detection systems, failure prediction), in healthcare (medical image analysis, such as tumor or brain damage detection), in image segmentation and in many other domains (Ghosal et al., 2020).

The word “clustering” doesn’t refer to a specific algorithm but rather to a general task that can be solved with a diverse set of methods which differ significantly in their notion of what constitutes a cluster. The definition of the latter, in fact, is not clearly defined beyond that of a group of objects having some similar property and this explains the great variety of algorithms currently present in the state-of-the-art. In general, every clustering algorithm distinguishes itself by following a different set of rules for defining the similarity among data points. The large number of different algorithms in existence can be grouped in a few categories, the most popular of which are the following:

- **Centroid models:** encompass algorithms for which the similarity between data points is measured relatively to their closeness to the clusters centroids, which are in turn calculated iteratively. These methods assume that the space is decomposed in \( k \) partitions which form a Voronoi tessellation. K-Means, probably one of the most well-known clustering algorithms, belongs to this category.

- **Connectivity models:** also known as hierarchical, they are based on the notion that data points closer to each other are more similar than the ones lying farther away. The clusters, which can be represented using a dendrogram, are usually formed either by assigning each data point to its own cluster and then aggregating the closest ones (“agglomerative clustering”), or by having all the data points initially belong to one big cluster and then partitioning it (“divisive clustering”).

- **Density models:** assume that denser areas of the data space should form separate clusters, so they iteratively search for such regions and try to connect the data
points. These methods have a wide spectrum of applications, but can have problems separating close clusters. One example of such algorithms is DBSCAN.

- **Distribution models:** assume that the data points will follow some statistical distribution (for example, Gaussian or Normal) and mark each data point with a probability of belonging to a certain cluster instead of another. These models are easily affected by overfitting.

Cluster analysis is a topic that falls under the umbrella category of unsupervised learning, which describes a set of machine learning techniques that aim at finding patterns and regularities in an unlabeled dataset. These methods have some advantages compared with supervised learning: the most immediate one is that there is no need to collect and label a large amount of data, which translates in an easier and faster training process. This advantage come with a cost: having no prior knowledge about the data they are manipulating, these algorithms offer less control over their results, which could consequently require some form of pre- or post-processing.

### 3.2.1.1 K-Means

K-Means is one of the most known clustering algorithms, belonging to the centroid-based category. Its name derives from the parameter $k$ which defines the number of clusters into which it partitions the $n$ data points, with the constraint that $1 < k < n$.

The algorithm initializes the position of the $k$ centroids, either using the Forgy method (which randomly chooses $k$ observations from the dataset) or by generating random starting coordinates. After that, K-Means begins an iterative process, the first step of which is known as the “assignment”: each data point is assigned to the closest cluster based on the Euclidean distance. Next, the “update” step recalculates the position of the centroids by considering the data points which were assigned to each cluster. This process continues until the algorithm converges on the final position of the centroids.

The major drawback of this approach to cluster analysis is that the number of clusters $k$ is an input parameter. This means that an inappropriate choice may lead to poor results and that the algorithm needs a certain amount of supervision by a human designer. In other words, it is unlikely that K-Means clustering will be able to perform adequately in the wild.
3.2. METHODS

3.2.1.2 X-Means

X-Means is a variation of the classical K-Means which was designed to overcome its strict dependency on having the value \( k \) provided as a parameter (Pelleg et al., 2000). This algorithm is able to select the optimal value automatically by performing model selection, which is the task of selecting a statistical model from a set of candidate models, given data. It does so by minimizing the Bayesian Information Criterion (BIC). The latter is defined as:

\[
BIC = k \ln(n) - 2 \ln(\hat{L}) 
\]  

(3.1)

Where \( k \) is the number of free parameters estimated, \( n \) is the number of data points and \( \hat{L} \) is the maximum value of the likelihood function of the model \( M \) with parameters \( \hat{\theta} \) for the data \( x \):

\[
\hat{L} = p(x | \hat{\theta}, M) 
\]  

(3.2)

This criterion can measure how effective is the parameterized model to describe the data and it penalizes the complexity, giving lower (better) scores to simpler models according to Occam’s razor heuristic.

X-Means operates by performing a K-Means operation with \( k \) set to its lower bound, then tries to split the centroids to obtain a higher number of clusters and continues until it is not possible to obtain a lower value of BIC.

3.2.1.3 Cluster Evaluation: Silhouette Analysis

A good clustering procedure has the objective to attain a high intra-cluster similarity and, at the same time, a low inter-cluster similarity, meaning that data points belonging to the same cluster should be highly similar between them and very dissimilar from the ones belonging to other groups. Several different methods are in existence to quantify the quality of a clustering operation, for example purity, which is based on assigning each cluster to the class which is more frequent between its data points (this approach has the disadvantage of needing a human supervision, as the data is unlabeled).

For the work we present in this thesis, we use an evaluation criterion known as Silhouette (Rousseeuw, 1987). This method provides a numerical value which is a measure of how similar an object is to its own cluster compared to other clusters and ranges from \(-1\) to \(+1\), with a high value representing a better score. If many data
points in a cluster have a high silhouette value, then the clustering is appropriate. This
index is calculated as:

\[ s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \]  \hspace{1cm} (3.3)

Where \( a(i) \) is the average distance between the data point \( i \) and all the other data
points in the cluster to which it belongs and \( b(i) \) is the minimum average distance from
\( i \) to all clusters to which it does not belong.

3.2.2 Principal Component Analysis

The centroid-based clustering techniques which we introduced in this section rely on
Euclidean distance as a metric for similarity between data points. This works well
for low dimensions, but in high-dimensional spaces the concept of distance between
two points becomes less precise, since it tends to converge. This happens because
as dimensionality increases, the volume of the space enlarges too and the data points
become more and more sparse. The problem is commonly known as the “curse of
dimensionality” (Bellman, 2013).

To avoid this problem, we have to rely on a series of techniques known as “dimen-
sionality reduction”, which as their name suggests aim to project high-dimensional
data into a lower-dimensional space preserving some meaningful properties of the orig-
inal data. One of the most common algorithms that perform this operation is Principal
Component Analysis (PCA), which computes a linear transformation between the two
representations (Hotelling, 1933). To reduce some data \( x \) from a \( n \)-dimension to a \( m \)-
dimension, with of course \( m < n \), it is necessary to find \( m \) vectors \( u^{(1)}, u^{(2)}, \ldots, u^{(m)} \)
onto which to project the data, so as to minimize the projection error. PCA works by
computing the covariance matrix of the original data:

\[ \sigma = \frac{1}{n} \sum_{i=1}^{n} (x^{(i)}) (x^{(i)})^T \] \hspace{1cm} (3.4)

and therefore calculating its eigenvectors through eigendecomposition and select-
ing the first \( m \) to form a reduction matrix \( U \). Finally, the projected data \( z \) is calculated as:

\[ z = U^T x \] \hspace{1cm} (3.5)

The number of dimensions \( m \) might be either chosen to be 2 or 3 for visualization
purposes or as the smallest value which preserves a certain amount of the variance (typically between 90-99%).

### 3.2.3 Hidden semi-Markov Models

In probability theory there is a class of stochastic models which are used to represent and elaborate predictions on randomly changing systems. Since these models comply with the Markov property (the assumption that a future state depends only on the current one with no influence from the past), these are known as Markov models. The Markov property is relevant and desirable since it allows reasoning that would otherwise be intractable.

One of the most common specializations of Markov models is the Hidden Markov Model (HMM), which assumes that the states being modeled are unobservable (hence the name “hidden”) and depend upon some other observable events. For example, the fairness of a coin which is being tossed can represent a hidden state which can be probabilistically deducted by observing the results of the flips, which are visible events. Mathematically, a HMM is defined by:

- A set of states $Q$.
- A transition probability matrix $A$ where each element $a_{ij}$ represents the probability of moving from state $i$ to state $j$.
- A set of observations $O$.
- An emission probability matrix $B$, where each element $b_{ij}$ represents the probability of an observation $o_i$ being generated by a state $j$.
- The initial probability distribution $\pi$ which models the probability for the Markov process to start in a certain state.

HMMs are used to solve three kinds of problems:

- **Likelihood**: given an HMM $\lambda = (A, B)$ and an observation sequence $O_s \in O$, determine the likelihood $P(O_s \mid \lambda)$. This is solved with a dynamic programming algorithm known as the “forward algorithm”.

- **Decoding**: given an observation sequence $O_s$ and an HMM $\lambda = (A, B)$, discover the best hidden state sequence $Q_s \in Q$ that generated those observations. The state-of-the-art algorithm to solve this problem is the Viterbi algorithm.
• **Learning**: given an observation sequence $O_s$ and the set of states $Q$, learn the parameters $A$ and $B$. The standard way to train an HMM is through the Baum-Welch, or forward-backward, algorithm.

For what concerns the work described in this chapter, we are mostly interested in the Viterbi algorithm. The latter is a dynamic programming method used to calculate the most likely path through the state transitions of an HMM over time. The need for this algorithm arises from the fact that, given $N$ states and $T$ observations, calculating the probabilities of all transitions would involve $N^T$ probability calculations, which is not computationally efficient. The intuition behind this algorithm is that it is not necessary to calculate all the transitions from all of the initial states to the final one, but it is sufficient to consider only the most efficient path from state $t_i$ to state $t_{i+1}$. When this is applied for each observation, it reduces the number of required calculations to $N^2T$.

HMMs are employed in a broad range of applications that span across different fields of science. For example, they are used to predict the intention of an agent for security reasons, as in anti-fraud or anti-malware systems (Alipour and Ansari, 2020), they are widely adopted by the bioinformatics community for genome sequencing and protein classification (De Fonzo et al., 2007) and are abundantly used in HRI for gesture recognition (Nguyen-Duc-Thanh et al., 2012), action learning by demonstration (Yang et al., 1997), emotion analysis (Kuhnlenz and Buss, 2004) and intention reading (Bien et al., 2005; Ognibene et al., 2019).

For the purpose of the research that we are about to present in this chapter, we are mostly interested in a variation on classical HMMs known as Hidden semi-Markov Model (HSMM) (Yu, 2010), illustrated in Figure 3.1. The main difference with the original model is that the probability of a state change depends not only on the observable events but also on the amount of time elapsed since the entry in the current state: a hidden state remains in the same state for a time duration $d$, emitting $d$ observations. Formally, this means that the definition of the model is enriched by the following:

- A duration matrix $D$, where each element $d_{ij}$ represents the probability of the state $i$ having the duration $j$.

HSMMs are used to solve the same three problems described in the context of HMMs and use variations of the standard algorithms which keep into account the increased complexity of the model. For example, the computational cost of the modified...
Figure 3.1: Basic structure of a HSMM. The hidden states (the goals) produce a sequence of visible observations (the human’s actions). Each state is moreover associated with a duration probability, which directly reflects the number of observations it will emit.

Viterbi algorithm increases proportionally to a factor equal to the longest possible duration in the model’s parameters intended as the number of discrete time-steps, $D_{\text{max}}$. The fields of application of HSMMs are similar to those of HMMs, including robotics: for example, they have been successfully used to teach manipulation tasks (Tanwani and Calinon, 2016) and for activity recognition by modeling multiple human walking events (Natarajan and Nevatia, 2007).

### 3.3 Model Description

Figure 3.2 offers an overview on the cognitive architecture we are about to introduce. As previously mentioned in Section 3.1, the latter is divided into a low-level action encoding paired with a high-level goal prediction. The system is designed to undergo a training phase, in which a teacher will demonstrate an open set of goals and the actions needed to achieve them, and an execution phase in which it will use this knowledge to infer the partner’s intentions as soon as possible.
CHAPTER 3. INTENTION READING

Figure 3.2: Overview of the intention reading architecture. The low-level extracts skeleton data from the optical stream, forms clusters and uses them to represent actions as transitions through clusters. The high-level uses this encoding to probabilistically infer the goal pursued by the partner. The robot control is in charge of performing the collaborative behavior.

3.3.1 Low-Level Skeleton Clustering and Action Representation

3.3.1.1 Skeleton Generation

The low-level module deals with human skeletal data generation and manipulation. The former is achieved through the use of a publicly available and pre-trained deep CNN architecture named OpenPose (Cao et al., 2016), customized for real-time multi-person 2D pose estimation. This neural computation receives in input 320x240 resolution images from the robot’s eye cameras and outputs a 18x2 feature vector representing the detected skeleton keypoints as 18 joints expressed in 2D spatial coordinates, as reported in Figure 3.3a. Because it has been proven that classification tasks achieve better results with a reduced set of joints (Manzi et al., 2017) and to optimize memory and speed requirements, we operate a keypoint reduction to diminish the volume of data required for each skeleton. To do so, we discard the keypoints corresponding to the eyes, the ears and the shoulders, whilst calculating a new torso keypoint as a median between the two hips: doing so, we obtain a more compact 11x2 representation shown in Figure 3.3b.

The skeletons generated by this procedure cannot be used directly for classification purposes, as they are dependent on the position and size of the subject. To overcome
3.3. MODEL DESCRIPTION

Figure 3.3: A comparison between the skeletal keypoints extracted from the camera image (a) and the reduced keypoint set computed by the system (b).

To address this problem and gain spacial invariance, we apply a normalization process first introduced by Cippitelli et al. (2016). For a skeleton with $n$ joints, the feature vector $f$ is defined as:

$$f = [J_1, J_2, ..., J_n]$$  \hspace{1cm} (3.6)

where $J_i$ is a vector containing the normalized 2D coordinates of the $i$th keypoint:

$$J_i = \frac{J_i - J_0}{\|J_i - J_0\|}$$  \hspace{1cm} (3.7)

where $J_0$ and $J_1$ are, respectively, the neck and torso joint. The latter will be located on the origin of the cartesian space, so its components will all be zero. For this reason, it is removed from the feature vector, which at this point will have a dimension of $10 \times 2$: this corresponds to a size reduction of approximately 44.5% compared to the original representation. This is important because the architecture is required to operate in real-time conditions.

### 3.3.1.2 Dynamical Clustering

Skeleton manipulation for action representation is done through an unsupervised clustering process. During the training phase, a teacher will demonstrate a set of goals and
CHAPTER 3. INTENTION READING

Figure 3.4: A diagram showing the training process of the low-level module. The image acquired by the robot’s eye camera is used to generate a keypoint representation of the human, which is then normalized before it can be clustered.

the actions required to achieve them. The robot will observe and detect, process and memorize skeletons from the video stream with an empirically chosen frequency of 2 fps. When the demonstration ends, the robot performs a clustering operation on the set of acquired skeletons. Each of the latter is represented in a high, 20-dimensional space which will subject the dataset to the curse of dimensionality, which we have discussed in Section 3.2.2. To overcome this problem, we apply PCA to perform a dimensionality reduction by projecting the 20D feature vectors on a 2D space. At this point, clustering can be performed and we employ X-Means as our algorithm of choice because of its ability to automatically estimate the best number of clusters in the data. To further justify our choice, we state that a collaborative intelligence operating in a truly unsupervised setting should be able to automatically adapt to its environment without requiring a direct human intervention, which would be the case when using clustering methodologies which require the explicit definition of the desired number of clusters, such as the more traditional K-Means. At the end of this process, the system will possess the 2D coordinates of the $n_c$ centroids and each skeleton will have been grouped to the closest one. Each cluster will represent a group of similar but not identical postures. The whole skeleton processing workflow is summarized in Figure 3.4.

Once the clusters are obtained, Transition Analysis is performed: each goal to be learned is encoded as a sequence of integers that represent the temporally ordered IDs of the clusters through which the demonstrator transitioned during the performance of the action, as shown in Figure 3.5. We impose an important restriction: we include
3.3. MODEL DESCRIPTION

Figure 3.5: Skeleton clustering. The observed skeletons are projected onto their 2 principal components \( \alpha \) and \( \beta \) and partitioned in similar postures. An action is defined as a particular transition, for example the goal “wave” is performed by a sequence of body postures that transition from cluster 3 to cluster 1 to cluster 2, hence the encoding \([3, 1, 2]\).

in the representation only the transitions through different clusters, discarding the persistence in the same group. This is done to make the final representation independent from the speed at which the action is performed, in fact gaining temporal invariance. Each action sequence is then associated to a goal label, which is requested verbally by the robot to the human. The combination of action encodings and associated goal names are forwarded to the high-level module to train it.

The execution phase, in which the robot has to actively infer an intention from its observations, relies on the data structures built during training: it receives a stream of camera images and for each of them it tries to detect a skeleton, process it as described in Section 3.3.1.1, project it on the reduced 2D space and associate it to the closest centroid. Once the cluster ID is obtained, it is immediately forwarded to the high-level module to allow it to formulate a prediction as quickly as possible.

3.3.2 High-Level Goal Probabilistic Inference

This module deals with probabilistic intention inference using the results of the low-level computation on the partner’s body posture: we need to predict the other agent’s
goals based on its actions. In other words, we are trying to infer a hidden, unobservable state from a series of observable events: this is a classic hidden Markov process. However, the application of a HMM is not possible because in this scenario each state can emit a sequence of observations and not just a single one (a goal will be formed from a series of low-level cluster transitions). To take this into account, we adopt a HSMM as the high-level computational model.

During the training of the system, the action representations and associated goal names learned from the low-level module are used to tune the parameter matrices of the HSMM. Normally this process is done through the Baum-Welch algorithm, but this procedure is generally adopted for large datasets which is not our case. Additionally, our formulation of the problem, and in particular on the structure of the transition matrix, simplifies our training conditions. For both these reasons we decide to follow a simpler approach in which we adopt a Maximum Likelihood Estimation (MLE) process. The $n$ action encodings received from the low-level form the observation sequences for the high-level, mathematically denoted as: $O_s = \{ o_s^{(1)}, o_s^{(2)}, ..., o_s^{(n)} \}$. Each encoding has a length $\text{len}(i) = \text{dim}(o_s^{(i)})$ for $i \in [1,n]$. Given this notation, the HSMM is parameterized as follows:

- The hidden states $Q$ are the goal names.
- The transition probability matrix $A$ is set to be uniform, because each intention can be observed at any moment in time with no particular causal relation.
- The observations $O$ are the range of possible cluster IDs.
- The emission probability matrix $B$ is calculated from $O_s$ by counting the frequency of every observation for each state in the training data and normalizing them to obtain values between 0 and 1.
- The initial probability distribution $\pi$ is set to uniform because we have no bias on the starting conditions.
- Finally, the duration matrix $D$ is defined by setting $d_{i,j} = 0.9$ for $j = \text{len}(i)$, while the remaining probability of 0.1 is shared between the other elements of the row.

During the execution phase, the high-level model receives in real time from the low-level a stream of cluster IDs observed in real time from the partner’s posture and applies the Viterbi algorithm to predict which hidden state $q$ is generating the observed events. This is a classic application of the HSMM decoding problem.
3.3. **MODEL DESCRIPTION**

This is, however, not sufficient: as we want the robot to act collaboratively before the action itself has ended, we are pursuing action prediction, not simply recognition. This means that we are trying to work with incomplete data, because we want to guess the hidden state before the sequence of observations is fully complete. To solve this problem, we support HSMM with an additional module which we call Anticipator: its role is to try to reconstruct in advance the missing observations, artificially generating enough data for the HSMM to look ahead of time and decode the goal correctly.

The process is carried out as follows: the high-level receives a cluster ID from the low-level, appends it in the observation sequence $O_s$ and immediately applies the Viterbi algorithm to calculate, for each event, which hidden state is the most likely to have generated that sequence. This computation yields a sequence of states $Q_s$ of equal length of $O_s$. Of course, the inference achieved with only one or two observations is not very reliable, so we accept a prediction only if it was formulated after observing 50% of the expected duration of an intention (for example, if we had a goal with an expected duration of 7, we would want to observe at least 3 symbols before accepting the prediction). Even so, we found that during the experimentation sessions the accuracy was not meeting our expectations because of the reasons explained above. The Anticipator supports the decisions of the HSMM by trying to match the sequence of partial observations to the action encodings used to train the probabilistic model: if an unambiguous match is possible, it will emit the remaining symbols of that sequence for the HSMM to perform a second inference, otherwise the previous one will be maintained. In other words, the Anticipator performs a look-ahead function that allows the HSMM to make a prediction using a complete set of observations reconstructed by a partial set of symbols.

The reason why the Anticipator is not used as a stand-alone predictor is that it is not able to do anything but match a sequence of symbols. If used without the HSMM, the system would become strongly affected by observation noise, such as a mis-classification of a skeleton by the low-level. By using this tool as a support for the HSMM, we still allow probabilistic computations that allows to reason under uncertainty.

As soon as one of the goals is confidently predicted, its label is forwarded to the robot controller to instantiate appropriate collaborative behavior.
3.3.3 Robot Control

The robot control module deals with the direct interface between the cognitive architecture and the robotic platform, namely an iCub robot. In particular, it provides interaction with its YARP middleware to control the sensors and actuators and perform vision, movement and grasping tasks for the shared goal task. The robot is able to see through the RGB cameras in its eyes, from which it can collect image frames that are used to generate skeletal data. It is also able to perform vocal communication due to its incorporated microphone and the connection to a set of remote speakers which transmit its synthetic voice generated through a text-to-speech software. In order to allow the robot to provide collaborative assistance during the experimental setup described in the next section, we have also programmed it to detect, grasp and hand to the human partner some objects that will be placed in front of it.

3.4 Experiment

The goal of this experiment is to validate the intention reading capabilities of the cognitive architecture on a shared goal scenario. In particular, it involves a collaborative block building game where the robot will help the human to assemble constructions using different colored toy blocks.

3.4.1 Experimental Setup

Figure 3.6 shows the experimental setup. The iCub robot and the experimenter are facing each other with a table in between. Some colored toy cubes are situated next to the robot: they are green to its right and blue to its left. A playing area, where the buildings will be assembled, is situated in between.

3.4.2 Procedure

The toy cubes can be used to build four different kind of structures: towers, walls, castles and stables, each of which consisting of three blocks. The first two may only be made of one single color each, $\gamma_1$ and $\gamma_2$, whilst the castle and the stable must be built using blocks from both groups. An example of these configurations can be seen in Figure 3.7. At the beginning of the experiment, iCub will not know the rules of the game and the experimenter will demonstrate all the intentions by performing
them one at a time and giving them a name. This interaction is mediated by vocal communication: the robot informs the human to be ready to observe and learn, he or she performs one of the goals and signals the end of the demonstration by pronouncing the word: “Stop”. At that point the robot will request a name to associate to what it just observed and will ask if the learning process is over or not. Once the human expresses no will to teach further goals, the cognitive agent uses the collected skeleton dataset (which is reported in Appendix B) to perform the training of the whole architecture, as described in Section 3.3. It is important to note that the experimenter can choose to demonstrate either \( \gamma_1 = blue \wedge \gamma_2 = green \) or \( \gamma_1 = green \wedge \gamma_2 = blue \).

During the execution phase, the human experimenter will start pursuing one of the goals and the robot will try to predict the intention as soon as possible and act in a helpful manner, that means fetching the remaining blocks needed to finalize the construction and hand them over to its partner, one at a time and in the correct order.

The color of the blocks is not relevant to the robot itself at this stage, meaning that what really counts is their position on the table which translates in a set of movements needed by the human to reach them. The color differentiation between the blocks serves as a visual indicator for the experimenter and to facilitate the comprehension of what is happening to the reader.

A video showing the experimental procedure is available online (Vinanzi, 2019a).
 CHAPTER 3. INTENTION READING

(a) Wall  (b) Tower  (c) Castle  (d) Stable

Figure 3.7: An example of the four structures taken into account in the intention reading block placing experiment. In this case, the color combination chosen was: \( \gamma_1 = \text{green} \land \gamma_2 = \text{blue} \).

Table 3.1: Trained experimental intentions: each goal is associated to a sequence of cluster transitions.

<table>
<thead>
<tr>
<th>Goal</th>
<th>Action representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>wall</td>
<td>[0, 2, 0, 2, 0, 2, 0]</td>
</tr>
<tr>
<td>tower</td>
<td>[0, 1, 0, 1, 0, 1, 0]</td>
</tr>
<tr>
<td>castle</td>
<td>[0, 2, 0, 1, 0, 2, 0]</td>
</tr>
<tr>
<td>stable</td>
<td>[0, 1, 0, 2, 0, 1, 0]</td>
</tr>
</tbody>
</table>

3.5 Results and Discussion

Figure 3.8 shows the resulting clustering after the training phase, performed by a single demonstrator: the learning data were autonomously divided in three clusters, each of which represents a particular pose of the human partner. In particular, cluster 0 collects skeletons in a central, standing position, cluster 1 groups skeletons leaning towards the robot’s right side while skeletons tilted to the robot’s left side fall under the domain of cluster number 2. To evaluate the quality of this unsupervised process we calculated the average Silhouette evaluation metric as the mean of the three Silhouette scores of the corresponding clusters. The result is \( \bar{S} = 0.858 \), which indicates that the number and placement of the centroids is adequate.

Each intention was then encoded as a series of transitions between the clusters: these ID sequences with the associated goal names are reported in Table 3.1. These sequences are easy to interpret, as they show the progression of the human partner from the neutral, central position to its left of right, back to the center and so on. To better understand them, Figure 3.9 visualizes the recorded skeletons that, during the teaching of the goal “wall”, have triggered cluster transitions. As noticeable from the skeleton
3.5. RESULTS AND DISCUSSION

Figure 3.8: The 2D projection of the resulting clustering obtained during the training phase of the block building game. Skeletons have been divided into three groups of similar postures and the numbered dots represent the centroids. Each data point shows the bounding box of the human partner as seen from the robot’s eyes, the position of which was fixed on the opposite side of the table.

IDs, which don’t progress linearly, the Transition Analysis method was successful in filtering out uninteresting data and provide temporal invariance to the encoding. It is also possible to notice that keypoints which are not visible by the robot, in this case the ones associated to the knees and feet of the human, are discarded by the computation. The decomposition of the remaining goals is reported in Appendix A.

During the execution phase of the experiment, we performed 80 games with the robot, 20 for each of the 4 goals. Since the transition probability matrix for the high-level HSMM is set to uniform, the ordering of the goals is not relevant. This stage was performed not by the same person who acted as a demonstrator, but rather from another one possessing different physical attributes, such as gender and height. This person was non-naïve with respect to the purpose of the experiment and had received training on the task to be performed.

The results we collected are summarized in Table 3.2. The latter shows, in order: the goal which was tested, number of trials performed, accuracy of the prediction, average number of cluster transitions encountered before formulating an hypothesis,
Table 3.2: Experimental results for the intention reading experiment. See Section 3.5 for the full explanation.

<table>
<thead>
<tr>
<th>Goal</th>
<th>Trials</th>
<th>Accuracy</th>
<th>Transitions</th>
<th>Time</th>
<th>Interception</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wall</td>
<td>20</td>
<td>100 %</td>
<td>4.00</td>
<td>4.41 s</td>
<td>57.14 %</td>
</tr>
<tr>
<td>Tower</td>
<td>20</td>
<td>100 %</td>
<td>4.05</td>
<td>3.69 s</td>
<td>57.86 %</td>
</tr>
<tr>
<td>Castle</td>
<td>20</td>
<td>100 %</td>
<td>4.00</td>
<td>4.63 s</td>
<td>57.14 %</td>
</tr>
<tr>
<td>Stable</td>
<td>20</td>
<td>100 %</td>
<td>4.05</td>
<td>5.22 s</td>
<td>57.86 %</td>
</tr>
<tr>
<td>Total</td>
<td>80</td>
<td>100 %</td>
<td>4.03</td>
<td>4.49 s</td>
<td>57.5 %</td>
</tr>
</tbody>
</table>

average time (in seconds) necessary for the inference and average percentage of completion of the action by the partner when the intention was predicted. We aimed at keeping this last value as low as possible, because we want the robot to be able to identify an intention and start acting collaboratively as soon as possible.

Note how the synergic combination of the HSMM and Anticipator has led to two major benefits: first of all, the intention prediction was perfectly accurate (these results compare well with the ones obtained from Manzi et al. (2017), who use a similar approach of skeleton encoding-decoding to perform activity recognition using a disembodied agent). Secondly, the prediction was fast enough to make the robot engage collaboratively when the partner had completed just more than half of the action (higher percentages of completion would diminish the usefulness of the collaborative robot, as the partner would already have completed most of the action without assistance).

3.6 Conclusions

In this chapter, we presented a novel artificial cognitive architecture to perform intention reading on a human partner engaged in a shared task. In particular, we have used unsupervised machine learning and Markovian probabilistic modeling to allow a humanoid robot to identify the goal pursued by its partner and to adopt a collaborative strategy. Following the psychology literature on intention reading development in children, this model is based on the experience that the robot acquires from its teachers and it divides the task into two levels of understanding, a mechanical and a more abstract one. The system has been validated on a human-robot interaction experiment consisting in a block building game where the iCub humanoid robot was initially instructed on the rules and, after that, was able to understand which goal was being pursued by its partner and act helpfully towards its achievement.

Many models in the current literature share the need for handcrafted plan libraries,
3.6. CONCLUSIONS

Figure 3.9: Decomposition of one of the intentions that are trained to the robot during the block building game, corresponding to the goal “wall”. The skeletons showed are the ones that triggered a cluster transition and show the visible upper body of the demonstrator manipulating blocks on the table.

limiting their capacities and flexibility. By using the developmental robotics approach, we managed to overcome this limitation: the set of goals that can be learned by the robot is open and, in the absence of ambiguities, at least theoretically unbounded. This means that the agent is able to learn an arbitrary set of goals and, if their required actions are different enough to be clustered separately, there is no theoretical upper limit to the amount of goals that can be learned. In addition to this, our approach proves to be robust, meaning that the low-level is both space and time invariant thanks to the normalization of the skeletons and the way in which Transition Analysis was implemented. Finally, unsupervised learning makes the robot learn on the fly with no need for big datasets or long training processes, making this architecture extremely lightweight from a computational point of view: the robot just has to observe each goal once to be able to identify it in the future. Other than that, despite this model being designed and tested for HRI, the way in which skeletons are detected from the visual field allows the use of this architecture even for humanoid-robot interaction scenarios, opening opportunities for new and different kind of studies.

The model that we presented here is to be considered as a starting point in our investigation of developmental plausible intention reading in embodied robotic agents.
As such, it presents some opportunities for future improvements which we wish to underline in order to inspire future research in this direction. During the description of the experiment in Section 3.4, we noted how the robot is not aware of the color of the blocks when trying to read the human’s intention. The spatial relationship between body posture sequencing and the objects in the scene is likely to offer interesting connections that might aid the robot in its task. These links could be explored, for example, through the use of qualitative space relation (QRS) descriptors. Other than that, there is psychological evidence that humans use gaze direction, other than body posture, to infer the intentions of another agent (Tomasello et al., 2005). The inclusion of multiple sources of information can lead the robot to be able to predict intentions in more complex scenarios which present a degree of ambiguity. We explore this direction in Chapter 4, in which we extend this architecture to take in account multimodal sensory perceptions.
Chapter 4

The Role of Social Cues for Goal Disambiguation

4.1 Introduction

The work we are about to introduce follows the one we have presented in Chapter 3 and represents an enhancement of our cognitive architecture for robotic intention reading. The aim of this follow-up is to allow the robot to perform its task in more complex scenarios, as mentioned in Section 3.6. In particular, we want to address the case in which the action/goal sequences that are taught to the robot present some degree of ambiguity, which can happen if the human demonstrator transitions between body configurations which don’t differ as much. For example, the skeleton data of someone touching their mouth might be similar enough to the one of someone touching their nose, meaning that these data points will likely be close to each other in the data space. This proximity would cause the clustering algorithm to encounter difficulties in separating them into different clusters and it will likely end up grouping them together, impairing the training and recognition of those specific actions.

To solve the ambiguity that arises in this situation, we have decided to enrich the representation of the human physical configuration: we state that by considering additional features other than the skeleton keypoints we are able to generate enough data to disambiguate what were previously considered similar postures. In other words, our idea is to make use of a multitude of different social cues to enrich the description of the human partner involved in joint activity. To achieve this result, the new architecture makes use of a novel multimodal clustering algorithm and is able to operate in scenarios where our first model would have struggled.
4.2 Methods

4.2.1 Multimodal Machine Learning

Typical machine learning algorithms operate on one type of input specific for their problem, for example on images or text. As the artificial intelligence revolution unraveled, researchers and scientists had the intuition that in order to achieve more complex tasks they should consider multiple different types of inputs to be analyzed at the same time, aiming at a more holistic view of the environment in which their systems were operating. This is especially true regarding robotics and in particular when considering social robots which operate in complex, human-shaped environments. Baltrušaitis et al. (2018) refer to the term “modality” as the way in which something happens or is experienced. In particular, they define sensory modality as one of the primary forms of sensation, such as vision or touch. Given this definition, we can classify machine learning algorithms as unimodal or multimodal based on the number of sensory modalities they keep into account. One of the earliest applications of multimodal machine learning was in the field of audio-visual speech recognition (AVSR), which tries to align voice audio signals and lips motion to achieve a better understanding of human speech using an extended version of HMMs (Dupont and Luettin, 2000).

Compared to classical machine learning problems, the multimodal approach offers the chance to gain deeper understanding from the environment but also offers some new challenges, specifically:

- **Representation**: multimodal data is often heterogeneous (as in the AVSR problem we described earlier) and needs special representations which are able to aggregate their different structures in a comprehensive description. There are two approaches which seem to be more common than others. In particular, joint representations rely on a function that operates on unimodal data:

\[ x = f(x_1, \ldots, x_m) \]  \hspace{1cm} (4.1)

while coordinated representations use projection functions that are able to map between the unimodal and multimodal space:

\[ f(x_1) \sim g(x_1) \]  \hspace{1cm} (4.2)
4.2. METHODS

- **Translation**: data needs to be mapped from one modality to another (for example, automated image captioning). This task is performed either by training dictionaries or using generative models.

- **Alignment**: is the problem of identifying related sequences between different modalities, such as matching a video with a script. This problem is solved by using some similarity metric between the modalities which can be either be explicitly defined by the designers of the system or implicitly calculated through unsupervised processes.

- **Fusion**: heterogeneous data must be eventually joined in order to formulate predictions, but different modes might have different predictive power. This problem can be overcome by either combining the data and then classifying it or vice versa, classifying the data from each modality separately and then fusing their outcomes.

- **Co-learning**: this challenge involves the exploitation of knowledge in one modality to aid the modeling of another one. This can be done by exploiting the parallelism of the dataset, if present, or by transfer learning in artificial neural networks.

4.2.2 Bayesian Networks

Bayesian networks (BNs) are a class of probabilistic graphical models, described through the use of directed acyclic graphs, which are used to model uncertain and complex domains in a broad range of applications. A BN is composed of the following:

- A set of nodes, each of them representing a random variable, which may be either continuous or discrete. For what concerns this thesis, we are going to consider only the latter.

- A set of directed edges that connect nodes such that an edge connecting node $A$ to node $B$ implies a principle of causality and a conditional dependence $P(B | A)$. In this case, $A$ is said to be a parent of $B$ and, vice versa, $B$ is a child of $A$.

- A probability distribution associated to each node $X$, which in case of discrete variables is a probability mass function that depends on its parents: $P(X | parents(X))$. The latter is usually represented with the use of a conditional probability table.
Figure 4.1: An example of Bayesian network with two nodes representing boolean random variables. The tables shown next to them are their respective conditional probabilities tables. Note how the size of the latter depends on the number of the parents of that node.

(CPT) that associates a particular set of values for the node’s parents variables to the probability of the variable represented by the node under consideration. In the case of a BN with Boolean variables, if a node has \( m \) parents the table will have \( 2^m \) entries.

An example of this probabilistic model is shown in Figure 4.1. BNs have two main advantages. First of all, their graphical representation through nodes and edges facilitates identifying the causal relationships between variables. The latter happens because these networks satisfy the local Markov property, which states that a node is conditionally independent of its non-descendants given its parents. This property allows us to simplify the calculations of joint distributions:

\[
P(X_1 = x_1, ..., X_n = x_n) := P(x_1, ..., x_n) = \prod_{i=1}^{n} P(x_i \mid \text{parents}(X_i))
\]  

(4.3)

In the equation above, we have denoted the random variables with capital letters and their possible values with small letters.

The second advantage is that, by using Bayes’s theorem they are able to calculate the probabilities of children nodes given their parents (that is, causes from consequences) but also the opposite (the probabilities of different causes given the consequences). In general, given an observation or belief on a node \( B \) it is possible to infer the posterior probability of a node \( A \) using the following equation:

\[
P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}
\]  

(4.4)
The simpler form of BNs is one in which there is only one single path between any two nodes in the network: these are called single connected networks. For these models, exact inference is linear in the size of the network, measured as the number of CPT entries. The inferential algorithm used for these models is known as Pearl’s Message-Passing, or belief propagation (Pearl, 1982). For multiple connected networks, it has been demonstrated that exact inference is NP-hard (Cooper, 1990) and it is only possible to obtain approximate results.

The use of BNs is widespread across a range of diverse domains. In robotics, they have been successfully used for a great variety of applications including but not limited to movement planning (Toussaint and Goerick, 2007), self-tracking using landmarks (Zhou et al., 2010) and estimating the intended destination of another agent (Escobedo et al., 2014).

4.2.2.1 Parameter learning

There are two ways in which the CPTs can be parameterized. The most intuitive one is the empirical way: the phenomenon which is being modeled is either analyzed by an expert or observed and measured, the collected values are then transformed into probabilities and these are finally recorded in the tables. This, of course, is not always possible and most of the time it would be preferable to train the network on some available data. If the network structure is not too complex, it is possible to adopt the Maximum Likelihood Expectation (MLE) approach: the latter consists of determining the values for the parameters so that they maximize the likelihood that the process described by the model produced the data that were actually observed (Aldrich et al., 1997). This process is known as “parameter learning” (Koller and Friedman, 2009).

In mathematical terms, given some data \( d \), the goal of MLE is to find the values of the model parameters \( \theta \) that maximize the likelihood function \( L(\theta; d) \) over the parameter space \( \Theta \):

\[
\hat{\theta} = \arg \max_{\theta \in \Theta} L(\theta; d)
\]  

(4.5)

In practice, it is more convenient to work with the log-likelihood, which is a logarithmic transformation of the likelihood function:

\[
\ell(\theta; d) = \ln L(\theta; d)
\]  

(4.6)

If \( \ell(\theta; d) \) is differentiable in \( \theta \), the necessary conditions for the occurrence of a
maximum are:

$$\frac{\partial \ell}{\partial \theta_1} = 0, \quad \frac{\partial \ell}{\partial \theta_2} = 0, \quad \ldots, \quad \frac{\partial \ell}{\partial \theta_k} = 0$$

(4.7)

The latter are known as the likelihood equations, whose approximate solution is found via numerical optimization.

### 4.3 Model Description

In this Section we present an artificial cognitive architecture that will enable a social robot to perform intention reading in HRI and that builds on the foundations of the model introduced in Chapter 3, scaling it to operate in more complex scenarios by considering multimodal perceptual inputs. This architecture is designed to be used in a collaborative context where the robot and a human partner are working together to reach a shared goal, the nature of which is not directly specified by the human and must be inferred by the artificial agent using non-verbal communication. The robot will use this model to observe its partner’s actions, intended as sequences of physical configurations, and perform probabilistic decision-making with the aim to provide appropriate assistance. The model we are about to introduce uses the same developmental robotics approach as the previous one.

The enhanced intention reading architecture closely follows the one schematized in Figure 3.2. The main difference is that we are now considering not only the skeleton keypoints configuration in our clustering operation, but also another social cue, in particular gaze direction. Our first attempt involved creating a joint representation $x = f(x_{\text{skeleton}}, x_{\text{gaze}})$ and processing it in the established workflow, that is attempting a clustering through X-Means and a Transition Analysis to obtain the action encodings. This trial was not successful, as the data points were not sufficiently sparse for the clustering algorithm to identify enough separate clusters and appropriately describe the different physical configurations. This led to the design and implementation of a new multimodal clustering algorithm: Feature-Space Split Clustering, which is described below.

#### 4.3.1 Feature-Space Split Clustering

We create action representations from the perceptual data through an unsupervised clustering procedure, using a novel algorithm that combines multiple sets of features
in several increasingly refined stages. This strategy is adopted because complex and potentially ambiguous actions can be distinguished by increasing the granularity of the clustering operation which keeps into consideration a multitude of social cues. Our method takes inspiration from a technique known as subspace clustering (Vidal, 2011), which consists in finding clusters that are defined by subsets of the available features, computed automatically, and by using these low-dimensional clusters to define higher-dimensional ones. The main idea behind Feature-Space Split Clustering (FSSC) is the one of a multilevel clustering process that uses only a subset of the features at each level.

Consider a set of $M$ training samples:

$$X = \{x^{(1)}, x^{(2)}, \ldots, x^{(M)}\}$$  \hspace{1cm} (4.8)

Each sample can be seen as defined by $N$ groups of features:

$$x^{(i)} = \{f_1^{(i)}, f_2^{(i)}, \ldots, f_N^{(i)}\}$$  \hspace{1cm} (4.9)

Each group defines the feature-space $f_n$ with $n \in [1, N]$ and contains data extracted from a different perceptual input. FSSC works by implicitly computing a tree of depth $L = N$ whose nodes contain the refined clusters. The root node ($\ell = 0$) contains all the data samples and is considered as a single cluster, while nodes of each subsequent level $\ell > 0$ are the clusters obtained by clustering the samples belonging to the parent node in the feature-space $f_{\ell}$. This process is illustrated in Figure 4.2. At each level, we perform Principal Component Analysis (PCA) dimensionality reduction to project the data in a 2D space to avoid the curse of dimensionality and we use X-Means as the internal clustering method. Algorithm 1 describes this computation.

Given the hierarchical and nonlinear nature of this algorithm, we can’t perform classification (intended as the association of a new data sample to one of the existing clusters) through a simple Euclidean distance search for the closest centroid. Instead, the procedure described in Algorithm 2 must be adopted. The latter searches through the cluster tree, comparing the centroids in their respective feature-space coordinates until a leaf node is found.

Similarly to subspace clustering (Vidal, 2011), FSSC finds dense regions in low dimensional spaces and combines them to form clusters. The main difference between the two algorithms is that the former performs feature selection whilst the latter uses the feature-spaces to divide a priori the features into groups.
Figure 4.2: A visual representation of the computation performed by FSSC. The samples at level 0 are clustered in the feature-space $f_1$ and produce several groupings at level 1. These are once again separately clustered in the feature-space $f_2$. The leaf nodes (in this example, at level 2) represent the final clusters.

We will now analyze FSSC from the perspective of the multimodal machine learning challenges which we introduced in Section 4.2.1. Following the taxonomy created by Baltrušaitis et al. (2018), the ones we have to address are: representation, fusion and co-learning. FSSC uses a joint representation, since each modality relies on the same function. The fusion process is conducted in an incremental way, since the algorithm processes one modality at a time, integrating its results with the ones collected until then. We do not explicitly address co-learning at this time as we don’t expect, for the research setting we are designing this algorithm for, to have a difference in resource richness between the different modalities. Finally, it is worth mentioning that even if this approach splits the feature-spaces, in fact it does not decouple the multimodal features extracted from the human, as they share a temporal dependence on the frame from which they were generated. In other words, the features never lose their alignment.

4.3.2 Low-Level Social Cues Clustering and Action Representation

Social cues are verbal or non-verbal signals that offer hints which facilitate social understanding, and as such are considered important communicative tools which can be
4.3. MODEL DESCRIPTION

Algorithm 1: Feature-Space Split Clustering (FSSC)

Input: training samples $X$; number of feature sets $N$
Output: A tree of clusters

$tree \leftarrow \{\}$

Initialize the root node with all the samples $X$

for $\ell \leftarrow 0$ to $N$

foreach cluster of level $\ell$ do

$x \leftarrow$ samples belonging to cluster

$f \leftarrow f^{(s)}_{\ell+1}$

$f' \leftarrow$ Dimensionality reduction on $f$

newClusters $\leftarrow$ CLUSTERING($f'$)

Set newClusters to level $\ell + 1$

$tree \leftarrow tree \cup$ newClusters

end

end

return $tree$

Algorithm 2: Cluster classification

Input: cluster tree $T$, testing sample $s$
Output: cluster to which $s$ belongs

$parentNode \leftarrow$ root node of $T$

$\ell \leftarrow 1$

Loop

$C \leftarrow$ descendants of $parentNode$ in $T$

cluster $\leftarrow$ min $distance(f^{(s)}_{\ell}, c \in C)$

if cluster has descendants then

$parentNode \leftarrow$ cluster

$\ell \leftarrow \ell + 1$

else

return cluster

end

EndLoop

used to share intentions (Freeth et al., 2013). Examples of social cues are body posture (the importance of which we have already examined), gestures, facial expressions from which it is possible to extract information about emotions, appearance and gaze direction.

We have decided which social cues to use in this study by taking inspiration from the findings of Tomasello et al. (2005) who state that the two main cues used by developing infants are biological motion and gaze. Both these social signals can be sourced
from the robot’s eye cameras, but despite them being both visual in nature they possess a different structure.

We generate skeleton data using the same process described in Section 3.3.1, by using the OpenPose library (Cao et al., 2016), reducing the verbose representation and then normalizing them to obtain spatial invariance. For what concerns gaze direction, we use Deepgaze (Patacchiola and Cangelosi, 2017), a CNN specialized in head pose estimation, to retrieve a 3D vector representing estimated roll, pitch and yaw for each image acquired by the robot. We chose to adopt head-gaze instead of eye-gaze to avoid some computational overheads that would impair the real-time computation of several frames per second. Despite evidence that eye tracking works better than simple head tracking in HRC (Palinko et al., 2016), this has been proved to be an acceptable approximation in settings that don’t require more detailed measures, such as the one that we are adopting for our experiment (Jha and Busso, 2017).

Given this premise, we operate FSSC with \( N = 2 \) and for each image \( i \) we have that \( f_1^{(i)} \) is a 20-dimensional vector containing the 10x2 skeleton keypoints configuration and \( f_2^{(i)} \) is a 3-dimensional vector that specifies the gaze direction.

During its initial training, the agent will observe the actions of its human partner, record the image frames and extract all the relevant features. This assembled dataset will be used to create the clusters using the method described in Algorithm 1. These will be used in the next stage, Transition Analysis, to create the low-level encoding: each action will be represented by the sequence of the leaf cluster ids encountered during its performance (this means that non-leaf cluster ids in the FSSC tree won’t appear in the final representation). Again, to obtain temporal invariance, we include in the encoding only transitions through different clusters, discarding the persistence in the same group. The results of this analysis, plus a set of unique names for each goal, are forwarded to the high-level module to train it.

After being trained, the system will be able to perform intention reading: the agent will observe its human partner during the execution of the action; each of their physical configurations will be classified to one of the known clusters using the procedure described in Algorithm 2 and the discovered ID will be forwarded to the high-level module for probabilistic inference.
4.3. MODEL DESCRIPTION

Figure 4.3: The Bayesian network used for high-level probabilistic goal prediction. The top node represents the intention of the observed partner, whilst the bottom node symbolize the observations (the action encoding symbols produced by the low-level module).

4.3.3 High-Level Goal Probabilistic Inference

The high-level module is in charge of goal probabilistic inference from the observed actions. What we are trying to achieve is not action recognition but rather prediction, this means that only one part of the action will be known and observable. Our objective is to determine the intention based on as few observations as possible, so that the robot will be able to contribute to the task before it is over.

Despite the combination of a HSMM and the Anticipator described in Section 3.3.2 working adequately, we were motivated to find a simpler, more efficient model which could provide the same functionality. Eventually, we decided for the adoption of a BN, the structure of which is shown in Figure 4.3. The top node denoted as $I$ represents the intention of the human partner and its probability distribution is equal across all the possible goals. The bottom nodes marked as $O_k$ with $k \in [1,K]$, where $K$ is the maximum length of the encoded actions, represent the observations. The values of these nodes span in the range of the possible leaf cluster IDs identified by the low-level module. The CPTs of the observation nodes are fitted from the training data provided by the low-level Transition Analysis (i.e. the action encoding associated to each goal name) using MLE. We assume that the probability of each observation depends on the
driving intention and by the precedent symbol encountered:

\[
P(O_1 \mid I) \\
P(O_k \mid O_{k-1}, I) : k \in [2, K] \tag{4.10}
\]

Once the probabilistic model is trained, it can be used for inference. During the execution phase, the robot will be observing the human and recording each cluster transition in real-time. The low-level module will forward these symbols to the high-level, which will treat them as sequential observations. Each time a new piece of evidence is added to the model, we use Pearl’s Message-Passing algorithm to calculate the marginal probability distribution for node \( I \) given the evidence. As soon as one of the goals is predicted with a probability greater than 0.5, it is sent forward along the processing chain to instantiate appropriate collaborative behavior. A higher threshold would lead to a more confident prediction but would also slow down the robot’s reaction time, potentially impairing the collaborative effort.

It is worth remembering that the adoption of a probabilistic model instead of a deterministic one, such as a lookup table, is justified by the potential noisiness of the observations. The classification performed by the low-level will not always be accurate and the agent is required to be robust against invalid input sequences.

### 4.3.4 Robot Control

The robot control module deals with the direct interface between the cognitive architecture and the robotic platform. Instead of an iCub robot, this architecture is connected to a Sawyer (Figure 4.4), not because the former was not operating as desired but to test the capacity of this architecture to adapt over multiple physical implementations. This means that the Robotic Control module had to be redesigned to operate not only with YARP but also with ROS, the most common and widespread open-source robotic middleware. We fetch images using the head camera and not the arm one as the latter would introduce new problems related to the different angles of view, since the limb is prone to movement. This robot is not equipped with microphones or speakers, so we decided to use an external microphone and to display its dialogue text on its body tablet. In order to use this robot during our experiments, we have also programmed it to detect, grasp and manipulate some objects which will be present in the experimental setup.
4.4 Experiment

The aim of this experiment is to validate the intention reading capabilities of the cognitive architecture on a shared goal scenario where the human partner displays ambiguous actions. It involves a collaborative block placing game where the robot will help the human to line up different colored toy blocks following a set of rules.

4.4.1 Experimental Setup and Procedure

The experimental setup is shown in Figure 4.5a. The robot and the human are facing each other on the two sides of a table. Four different colored blocks are positioned on the corners of the playing area; anticlockwise from the top left they are: blue (B), orange (O), red (R) and green (G). The central area of the table is denoted as the building space.

The aim of the game is to use the available blocks to form a line, following a simple rule only known by the demonstrator: the blocks must be chosen one by one from a different side of the table (left or right). The 8 legal combinations of blocks are reported in Figure 4.5b and each of them forms a goal for our intention reading purposes. During the training phase, the human will demonstrate each goal once and the robot will form an action encoding as described in Section 4.3. An example of one of these demonstrations is provided in Appendix B. The goal name is determined
CHAPTER 4. SOCIAL CUES FOR GOAL DISAMBIGUATION

Figure 4.5: Experimental setup for the block building game. (a) Schematic of the playing table, depicting the position of the 4 colored blocks: blue (B), orange (O), red (R) and green (G). (b) The 8 admissible block sequences obtained by picking blocks alternatively from each side. These sequences are the goals for this scenario.

automatically by the robot through observation of the building space: the artificial agent will automatically form a description based on the colors of the blocks. To form a description which will facilitate human interpretation, the robot will scan the construction from the left side of the table (seen as above in Figure 4.5a) to the right. For example, if the human has lined up the sequence blue, green, orange and red, the robot will name it “BGOR”. It is worth to emphasize that the robot is not learning the underlying rule, but only the valid combinations.

During the execution phase, the human will choose one of the goals and start placing blocks accordingly. The robot will have to predict as quickly as possible which of the learned goals is being pursued and act collaboratively by collecting one at a time the remaining blocks and placing them in the building space to complete the construction.

4.4.2 Training Post-processing

To ground the general architecture to our specific experimental setup, we introduce a few constraints to the learning process. First of all, we instruct FSSC to consider the
biggest cluster resulting from the first clustering process in $f_1$ as the “neutral” cluster, the one in which the human is either standing straight in front of the robot or placing a block in the construction area. This cluster, which we will call $N_c$, is excluded from the remaining clustering computations, as it does not require any kind of refinement.

The second constraint which we add stems from the consideration that the system is sensitive to noise because it is learning each intention and its sequence of actions from a single training example, which is in turn obtained from a non-deterministic and unsupervised process. To reduce this effect, we implement a post-processing computation that aims to capture the regularities of the data, assigning to each end position of our actions (i.e. the grasping position for each of the blocks) one of the cluster IDs based on its statistical mode in the computed training dataset.

This is how we operate, in detail: once all the goals have been enacted once by the teacher, the robot executes its training process and obtains the whole set of action encodings and corresponding goal names. Each goal $g$ will be autonomously composed of a sequence of 4 IDs (one for each block to collect and place), each of them preceded and succeeded by $N_c$, as such:

$$g = [N_c, id_1, N_c, id_2, N_c, id_3, N_c, id_4, N_c] \hspace{1cm} (4.11)$$

We can eliminate $N_c$ and reduce this representation to the following:

$$g = [id_1, id_2, id_3, id_4] \hspace{1cm} (4.12)$$

each of those IDs should correspond to the respective block in the sequence. For example, if our goal was “BGOR” we would expect our representation to have the following meaning:

$$BGOR = [B = id_1, G = id_2, O = id_3, R = id_4] \hspace{1cm} (4.13)$$

Because of potential noise in the training data, this is not always the case. For this reason, we assign each end position to the statistical mode of their IDs in the whole training set. For example, when considering the corresponding cluster ID for the human reaching for block B, we ask each of the 8 action encodings which ID they have assigned to that position and we evaluate where is the general consensus. We repeat this process for all of the 4 end positions B, O, R and G and finally we substitute these calculated values into the action encodings. This limits the chances to have any errors in the training set. Figure 4.6 shows this process applied to a partial, example
Figure 4.6: An example of the post-processing operation which is performed on the training set. Suppose we are considering only 4 out of the 8 goals, the aim is to correct any irregularities in the data. (a) A possible partial training set that associates a goal label with a sequence of cluster IDs. The last value of GORB is incorrect due to a noisy observation. (b) The statistical mode for the end position B is calculated across the partial dataset: 3 out of 4 entries have assigned it to cluster ID 3. (c) The correct value is substituted in the dataset, fixing the error and allowing a correct training of the system.

Table 4.1: Action encodings for the experiment on goal disambiguation. Each goal is associated to a sequence of cluster transitions.

<table>
<thead>
<tr>
<th>Goal</th>
<th>Action representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGOR</td>
<td>[1, 4, 1, 6, 1, 5, 1, 7, 1]</td>
</tr>
<tr>
<td>BROG</td>
<td>[1, 4, 1, 7, 1, 5, 1, 6, 1]</td>
</tr>
<tr>
<td>GBRO</td>
<td>[1, 6, 1, 4, 1, 7, 1, 5, 1]</td>
</tr>
<tr>
<td>GORB</td>
<td>[1, 6, 1, 5, 1, 7, 1, 4, 1]</td>
</tr>
<tr>
<td>OGBR</td>
<td>[1, 5, 1, 6, 1, 4, 1, 7, 1]</td>
</tr>
<tr>
<td>ORBG</td>
<td>[1, 5, 1, 7, 1, 4, 1, 6, 1]</td>
</tr>
<tr>
<td>RBGO</td>
<td>[1, 7, 1, 4, 1, 6, 1, 5, 1]</td>
</tr>
<tr>
<td>ROGB</td>
<td>[1, 7, 1, 5, 1, 6, 1, 4, 1]</td>
</tr>
</tbody>
</table>

dataset. The new action encodings, plus the goal names, are forwarded to the high-level to train it.

4.5 Results and Discussion

After being trained to recognize the 8 goals referenced in Figure 4.5b, obtaining the action encodings presented in Table 4.1, the cluster tree generated by FSSC is reported in Figure 4.7. The axes of each graph are the two Principal Components in which the feature-spaces have been projected during the dimensionality reduction process.

During the first step of computation (Figure 4.7a), all the training samples have been clustered in the $f_1$ feature-space, which contains the skeleton keypoints of the
### 4.5. RESULTS AND DISCUSSION

Table 4.2: Experimental results for the goal disambiguation experiment. See Section 4.5 for the full explanation.

<table>
<thead>
<tr>
<th>Goal</th>
<th>Accuracy</th>
<th>Time</th>
<th>Intercept</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGOR</td>
<td>0.80</td>
<td>5.65</td>
<td>51%</td>
<td>0.80</td>
<td>0.64</td>
<td>0.71</td>
</tr>
<tr>
<td>BROG</td>
<td>0.70</td>
<td>6.07</td>
<td>53%</td>
<td>0.70</td>
<td>0.74</td>
<td>0.72</td>
</tr>
<tr>
<td>GBRO</td>
<td>0.90</td>
<td>6.53</td>
<td>51%</td>
<td>0.90</td>
<td>0.86</td>
<td>0.88</td>
</tr>
<tr>
<td>GORB</td>
<td>0.90</td>
<td>5.98</td>
<td>50%</td>
<td>0.90</td>
<td>0.95</td>
<td>0.92</td>
</tr>
<tr>
<td>OGBR</td>
<td>0.75</td>
<td>6.36</td>
<td>50%</td>
<td>0.75</td>
<td>0.88</td>
<td>0.81</td>
</tr>
<tr>
<td>ORBG</td>
<td>0.85</td>
<td>4.96</td>
<td>50%</td>
<td>0.85</td>
<td>0.89</td>
<td>0.87</td>
</tr>
<tr>
<td>RBGO</td>
<td>0.80</td>
<td>6.02</td>
<td>50%</td>
<td>0.80</td>
<td>0.70</td>
<td>0.74</td>
</tr>
<tr>
<td>ROGB</td>
<td>0.70</td>
<td>6.66</td>
<td>50%</td>
<td>0.70</td>
<td>0.82</td>
<td>0.76</td>
</tr>
<tr>
<td>Mean</td>
<td>0.80</td>
<td>6.03</td>
<td>51%</td>
<td>0.80</td>
<td>0.81</td>
<td>0.80</td>
</tr>
</tbody>
</table>

The partner’s body. This operation generated 3 clusters: a neutral one (ID 1), where the demonstrator was approximately standing straight, and the other two (IDs 2 and 3) containing skeletons leaning respectively left and right. The average Silhouette evaluation metric for this clustering is $S = 0.71$, which indicates a good partitioning.

In the second stage of processing (Figures 4.7b and 4.7c), clusters 2 and 3 are independently clustered: the samples contained in their domain are clustered in the $f_2$ space, expressing the gaze direction. Both these clusters are split in other two groups that capture the interaction with the far block (B or G) or the near one (O or R). The Silhouette scores for these two splittings are $S' = 0.86$ and $S'' = 0.69$.

It is now possible to see that the two stages of progressive clustering have divided the space of physical configurations of the human partner in 5 partitions: left/right and near/far, which correspond to the position of the 4 colored blocks, plus the neutral position. The adoption of multiple social cues has led to a granularity that would have not been possible by using only one set of features, say the skeleton keypoints. The adoption of a multimodal perception has resulted in a finer division of space which was able to correctly disambiguate the actions and therefore the goals.

During the execution phase of the experiment, we run 20 trials for each goal, totaling 160 games where the robot had to guess the current intention in progress. The results of this experimentation are reported in Table 4.2. For each goal, the latter displays: the average accuracy across 20 trials, the observation time measured as number of seconds needed to perform a prediction, the percentage of completion of the action by the partner when the intention was predicted, average precision, recall and F1 score. The last line of the table reports the mean of these scores across all goals. The robot
was able to perform correct intention reading approximately 80% of the time, identifying the pursued goal when its partner was halfway along its completion, giving it enough time to act cooperatively.

Finally, we report two confusion matrices: one for the overall goals (Figure 4.8) and one for the single blocks (Figure 4.9). Analyzing the latter, it is interesting to note that the system never confused two blocks belonging to different sides of the table, rather it tends to sometimes misclassify the near and far blocks of each side. This was an expected outcome, as those are the actions that present the most commonalities and ambiguity.

In comparison to our previous work, we have obtained less overall accuracy but we were successful in performing intention reading in a more complex scenario: with respect to our past setting, we have doubled the number of goals and partially overlapped some of the actions, creating a more error-prone environment that our previous methodology would have struggled to grasp.
Figure 4.7: Clustering results. (a) First level of FSSC in $f_1$ feature-space (skeleton keypoints) of all the training samples. (b) Second level of FSSC in $f_2$ feature-space (gaze direction) on cluster 2 and (c) cluster 3. Refer to Section 4.5 for the full details.
CHAPTER 4. SOCIAL CUES FOR GOAL DISAMBIGUATION

Figure 4.8: Confusion matrix for the 8 goals of the block placing experiment.

Figure 4.9: Confusion matrix for each individual block across the whole 160 experimental trials.
4.6 Conclusions

In this chapter we introduced a new multimodal clustering algorithm to combine several sensory inputs and we used its outcome to define a low-level encoding of the observed actions, which are then translated to high-level probabilities for each of the learned goals. We have tested this model on a block placing game where a Sawyer robot must try and help its human partner to complete a construction, demonstrating that the adoption of multiple social cues leads to a better goal disambiguation.

This piece of work represents an incremental step from our previous experiment on intention reading. In the latter, the general structure of the cognitive architecture was the same but the operations performed at each stage were simpler, as the experimental setup presented actions easier to represent and quite distinct from each other. In this chapter, we have introduced a more complex scenario that presents a degree of ambiguity between several actions and a wider goal space: this has led us to improve our methods with the addition of the FSSC algorithm and a finer processing of the data at each stage of execution. The results we have obtained are in line with our expectations: the ambiguity of the actions has decreased the overall performance of the robot, but the new methodology made it possible to still correctly read the intention on average 80% of the time.

Of course, even this refinement to our previous model leaves space for further possible improvements. We identify two areas where to address future studies: the first one is the addition of hierarchical intentions, in other words goals composed by multiple sub-goals (for example, uncapping a bottle might be one step to achieve the “drink” goal). This could be done by realizing an architecture with multiple high-level modules organized hierarchically so that a second-order BN would accept as observations the results of a first-order network. Another possible study could explore the use of more social clues and the investigation of their order of application within FSSC.
Chapter 5

Trust and Theory of Mind

5.1 Introduction

The technological revolution taking place in the fields of robotics and artificial intelligence seems to indicate a future shift in our human-centered social paradigm towards a greater inclusion of artificial cognitive agents in our everyday environments. This means that collaborative scenarios between humans and robots will become more frequent and will have a deeper impact on everyday life. In this setting, research regarding trust in HRI assumes a major importance in order to ensure the highest quality of the interaction itself, as trust directly affects the willingness of people to accept information produced by a robot and to cooperate with it. Many studies have already explored trust that humans give to robots and how this can be enhanced by tuning both the design and the behavior of the machine, but not so much research has focused on the opposite scenario, that is the trust that artificial agents can assign to people. Despite this, the latter is a critical factor in joint tasks where humans and robots depend on each other’s effort to achieve a shared goal: whereas a robot can fail, so can a person. For an artificial agent to know when to trust or distrust somebody and adapt its plans to this prediction can make the difference in the success or failure of the task.

The work we present in this chapter is centered on the design and development of an artificial cognitive architecture for a humanoid autonomous robot that incorporates ToM and episodic memory, as we believe these are the two key factors for the purpose of estimating the trustworthiness of others. We have based our architecture on an established developmental psychology experiment on ToM maturity in children of different ages and we were able to reproduce the same results on our robot, thus confirming that our approach successfully models trust mechanisms and dynamics in cognitive robots.
5.2 Background

In the pursuit of the developmental robotics approach, our aim for this investigation was to formalize and implement an artificial cognitive architecture able to reproduce the mechanisms that lead human children to estimate the trustworthiness of the agents they interact with. To do so, we decided to base our research on a psychology experiment by Vanderbilt et al. (2011) on ToM maturity in children. The authors of that publication have conducted several experiments, but the one which is relevant for our purposes is the first one described in their paper: 90 preschool-age children, equally divided in 3-, 4- and 5-years-olds, were shown a video in which an adult actor, either a helper or a tricker, gave advice to another adult that was trying to locate a sticker hidden in one of two boxes. Helpers would suggest the correct location, whilst trickers always pointed to the wrong box. The subject would follow their indications and search in the suggested box, either finding the prize or not. The children were given time to familiarize with those informants and their behaviors and were subsequently subjected to the test themselves with the same informants (Figure 5.1). This time, the child did not have to blindly follow the instructions he or she received but was free to decide if to trust or not the informant based on the past behavioral observations. Based on the children’s choices and on some meta-cognitive questions submitted to them, the authors of that study stated that only the 5-year-olds were able to effectively differentiate the helpers from the trickers, therefore demonstrating to possess a mature ToM.
which allowed them to simulate the mental life of other people.

In order to substitute one of the preschoolers with a humanoid robot, the latter needs to possess a trust and ToM computational model that allows it to predict the intentions and beliefs of the pointers it is going to interact with. A good candidate for this model is the developmental Bayesian model of trust designed by Patacchiola and Cangelosi (2016) which uses a probabilistic approach to solve the problem of trust estimation. This particular model uses discrete Boolean variables that assume two states: $a$ and $b$, each corresponding to one of the two positions where the stickers can be located in the experiment. A graphical illustration of this Bayesian Network (BN) can be observed in Figure 5.2: the two nodes $X_R$ and $Y_R$ represent respectively the beliefs and actions of the robot. The posterior distribution of the node $Y_R$ allows the agent to choose the action to perform: that means searching for the sticker in position $a$ or $b$. The connection between $Y_I$ and $Y_R$ represents the influence that the opinions of the informant have on the agent’s action. The action of the agent is then a consequence of its own belief $X_R$ and the informant action $Y_I$. Lastly, the estimation of $X_I$, the informant’s belief, makes the agent able to effectively discriminate a trickery from a non-malevolent human error. The cognitive architecture we designed creates one of these BNs for each informant it interacts with and uses it to predict their future behavior.
Figure 5.3: Overview of the trust estimation architecture. The robot uses its sensors to identify each human and select their own BN or eventually generate a new one using episodic memory. The selected model is then used for inference.

5.3 Model Description

Figure 5.3 presents an overview of our cognitive architecture for robotic trust estimation. This system is designed to interact with a multitude of informants, for each of which the robot will create some experiential memories which will be used in the process of evaluating their trustworthiness.

5.3.1 Perception

This module is directly connected with the robotic hardware and allows the robot to interface itself with the environment and the humans with which it will be interacting. Since this experiment didn’t require any object manipulation but rather a more social presence we decided to use a Pepper robot as our platform of choice. Compared to the other robots we have used in our previous experiments, this one offers a set of functionalities provided by its NAOqi framework which facilitate the preparation of our experimental scenario and offer easy access to its sensory perceptions. In particular, we required the robot to be able to synthesize vocal outputs to guide the users through the course of the experiment and to process vocal commands, but we also make an
extensive use of computer vision techniques, which use a mixture of programming libraries native on the robot’s operating system and OpenCV (Culjak et al., 2012). During the informant analysis, the robot will execute a face tracking task to lock onto the nearest human in its field of view, following his or her movements with the head. At this point, video images are captured from the frontal camera at a frequency of 15 fps and at a resolution of 320x240 pixels in RGB color space. Each of these frames are then processed for face detection, which is performed via the Haar Cascade algorithm (Viola and Jones, 2001): this consists of a machine learning approach where a series of classifiers of progressively greater complexity are applied in sequence to a sliding window inside an image, until one of them fails or all of them are passed, in which case the window is considered a match. This assures that the greatest computational-expensive tests are executed lastly and only when many other less resource-intensive ones have already passed. This algorithm will detect multiple faces present in the scene, if any, but only the one with the widest bounding box, that means the nearest to the robot, will be considered. Subsequently, the face is cropped, converted to greyscale and resized to 64x64 pixels, then added to an initially blank dataset. The latter will serve to train a second machine learning component, this time used for face recognition and based on Local Binary Pattern Histogram (Ojala et al., 2002). This algorithm works by dividing the image in cells and, for every cell, comparing the pixel value with each of the 8 nearest neighbors: whenever the center has a greater value, that neighbor will be considered a 0, otherwise a 1. Collecting these values in a clockwise manner will lead to the composition of a 8-bit binary value that is subsequently converted in a decimal number. After all the cells have been computed in this way, a histogram of these values is created and used as a feature vector for classification purposes. The output is an ID representing the face recognized, or -1 if no match is found. For better performance, our modified algorithm formulates predictions on 5 different face frames and selects as output the most predicted ID. In addition to informant identification, the vision module provides the sticker detection functionality by means of the native APIs of the robot’s operating system.

5.3.2 Trust Modeling

To be able to infer the trustworthiness of an informant, a robot has to both identify him or her, following the workflow presented in Section 5.3.1, and create a BN such as the one described in Section 5.2. The parameters of the latter will then be tuned through experience, as the robot familiarizes with its partner’s behavioral patterns.
5.3. MODEL DESCRIPTION

We define an “episode” as a data structure that encodes sticker searching events inclusive of both the sticker position and the suggestion received from the informer:

$$\mathbf{e} = [X_r, Y_r, X_i, Y_i]$$ (5.1)

The way by which this data is created depends on the agent’s ToM matureness: in case of a misleading suggestion, the immature agent associates the action $$Y_I$$ to the wrong belief $$X_I$$, whereas the agent with mature ToM identifies the deception and recognizes that $$Y_I = \neg X_I$$. Because of this deficit in reading the informant’s intention, the agent with immature ToM collects wrong statistical data that will distort inference in subsequent phases.

Once the agent has collected a certain amount of episodes from an informant, it can generate a BN associated to him or her using MLE to determine the CPTs of its nodes. Given the simplicity of the structure of this network, for the root nodes $$X_I$$ and $$X_R$$ we calculate these probabilities as:

$$P_Y(a) = \theta$$
$$P_Y(b) = 1 - \theta$$ (5.2)

Denoting $$N_a$$ and $$N_b$$ as the number of times the pointer chooses $$a$$ or $$b$$, we can estimate $$\theta$$ as:

$$\hat{\theta} = \frac{N_a}{N_a + N_b}$$ (5.3)

The CPT values for nodes $$Y_I$$ and $$Y_R$$ are calculated similarly, but have to take into account the higher number of parameters given from the conditional dependence on their parents.

Once a BN has been created for a certain user and its parameters have been learned from the interactions, it is possible to infer the posterior probability of the nodes given some observations. In particular, we are interested in estimating the belief given an action and vice versa. We calculate posterior distributions using Pearl’s Message-Passing algorithm, as described in Section 4.2.2.
5.3.3 Episodic Memory

The ability to use one’s own past memories to take decisions in the present and future is an important ability that enhances the cognitive processes. An implementation of this skill would enable the robot to react reasonably towards novel informants with which it has never familiarized. On a technical level, the main problem is to generate on the fly a new BN with adequate parameters to use with that unknown informer. These parameters will depend upon the robot’s personal character built in respect of the way it has been treated in the past: an agent which has been tricked often would learn to be mistrustful and vice versa, as in the “trust vs mistrust” phase in child development (Erikson, 1993), as discussed in Section 2.4.

The design guidelines that we followed in the creation of our algorithm were the following: memories fade away with time, the details become blurred proportionally to the amount of memories obtained and, finally, shocking events such as surprises and betrayals should be more difficult to forget than ordinary, expected ones.

Our algorithm draws inspiration from the particle filter technique widely used in mobile robot localization (Rekleitis, 2004). Whenever an unknown informant is met, this component generates on the fly a certain number of episodes to train a new BN.

We define the set of BNs memorized by the agent as:

\[ S = [s_0, s_1, \ldots, s_n] \]  

(5.4)

Where \( n \) is the number of informants known by the agent. Each BN \( s_i \) was generated by a set of episodes, and these are going to be denoted as “replay datasets” for that BN:

\[ E_{s_i} = [\epsilon_0^{(s_i)}, \epsilon_1^{(s_i)}, \ldots, \epsilon_m^{(s_i)}] : s_i \in S \]  

(5.5)

Where \( m \) is equal to the number of episodes of the replay dataset. So, in this notation \( \epsilon_j^{(s_i)} \) represents the \( j \)-th episode of the replay dataset that formed the BN \( s_i \).

The equation we are about to introduce uses information theory to quantify the amount of information each specific episode represents. Our goal is to find how much this value differs from the total entropy of its replay dataset: a high difference means that the event is to be considered surprising and must be easier to recall than ordinary, unsurprising events. For example, if an informant who is always been trustful suddenly tricks the agent, this betrayal will be remembered with a greater impact. At the same time, all of the memories are subject to a progressive time degradation that tends to
blur them with a timing dependent on their importance.

Formally, a real factor denoted as importance value \( v \) defined in the interval \([0, 1]\) is calculated for every episode \( \varepsilon_j^{(s_i)} \) as the difference between the amount of information of the episode, \( I(\varepsilon_j^{(s_i)}) \), and the total entropy of its replay dataset, \( H(E_{s_i}) \), divided by the discrete temporal difference from the time when the memory was formed.

\[
v(\varepsilon_j^{(s_i)}) = \frac{|I(\varepsilon_j^{(s_i)}) - H(E_{s_i})|}{\Delta t + 1} = \frac{|-\log_2 P(\varepsilon_j^{(s_i)}) + \sum_{\varepsilon \in E_{s_i}} P(\varepsilon) \log_2 P(\varepsilon)|}{t_{\text{present}} - \sum_{\varepsilon_j^{(s_i)}} + 1}
\]

(5.6)

Once \( v \) has been calculated it can be used to determine a replication factor by projecting it on a step function defined as:

\[
F(v) = \begin{cases} 
0 & \text{if } 0 \leq v \leq 0.005 \\
1 & \text{if } 0.005 < v \leq 0.3 \\
2 & \text{if } 0.3 < v \leq 0.6 \\
3 & \text{if } 0.6 < v \leq 1 
\end{cases}
\]

(5.7)

Every episode from each replay dataset in the agent’s memory is replicated \( F(v) \) times.

The thresholds of the step function have been defined by observing the probability distribution of the importance value \( v \) obtained while making the robot interact with five different informants, the latter characterized by the following percentages of helpful interactions: 100\%, 75\%, 50\%, 25\% and 0\%. This distribution is observable in Figure 5.4. Most samples tend to fall in the range \([0, 0.3]\), so samples that are contained in this interval are kept as they are, with no replications. Samples included in the intervals \((0.3, 0.6]\) and \((0.6, 1]\) are the most surprising ones for the agent and get, respectively, duplicated and triplicated. The bottom 5\% of the domain, i.e. \( v \) values less or equals than 0.005, are discarded (they are forgotten).

The next step in our algorithm is to pick \( k \) episodes to form the replay dataset for the new BN we intend to create, \( E_{s_{n+1}} \). It is possible to select random samples but this will lead to poor final results, so we instead operate a systematic resampling (Douc and Cappé, 2005), a kind of weighted random sampling. To avoid biases dependent on the two positions in which the sticker can be located, we want the agent to distinguish only
between positive and negative actions, or “truth” and “lies”: for this reason, instead of picking $k$ samples the system will only select $k/2$ and for each of them it will generate the corresponding symmetric example. For instance, if a \{sticker in a, correct suggestion\} episode is sampled, a \{sticker in b, correct suggestion\} episode will also be included in the new replay dataset.

The optimal number of samples $k$ has been investigated: a low value would result in a gullible belief network, whilst a high value would make it stubborn to changes. Our goal was to let the robot possess a firm but changeable prejudice about the novel informant. By analyzing the mean entropy of the episodic networks generated by different values of $k$, as showed in Figure 5.5, we selected $k = 10$ for the following reasons: it is an even number, so it will produce an integer $k/2$ value, it is neither too low or too high to incur in the above mentioned problems and, finally, it is a local minimum of entropy.

Finally, MLE is applied to the new replay dataset to evaluate the parameters of the network. This new BN will be stored in the agent’s long term memory as $s_{n+1}$ and will be used to predict the trustworthiness of the new informant.
Figure 5.5: Mean entropy of episodic memory networks generated with varying number of samples. The error bars represent the standard deviation of that batch. Given the random component intrinsic in the algorithm, a very large number of samples \(10^5\) have been generated for every value of \(k\).

### 5.3.4 Evaluation criteria for belief networks

We introduce the trust factor \(T\) as a measure of how keen is a BN to trust the informant.

To calculate \(T\), we hypothetically imagine that the sticker is located in position \(a\) and execute a belief estimation task, as we will describe in Section 5.4.2.3, to obtain the posterior probability \(p\) of node \(Y_I\): what we are doing is computing the probability that the informant will give correct advice given the matureness of the agent’s ToM. At this point, the value is scaled in a \([-1, +1]\) interval, where the two endpoints \(-1\) and \(+1\) represent, respectively, complete trickers and helpers, that means BNs whose parameters have been computed from replay datasets containing only truthful or untruthful episodes. Values of \(T\) in between represent informants that are partially helpful and misleading. To perform the scaling we required the maximum and minimum values of \(p\) and we found them by building two BNs formed by a large \(10^4\) number of, respectively, helpful and misleading episodes and computed \(p\) for each of them, resulting in \(p = 0.75\) for the helper network and \(p = 0.25\) for the tricker one.
For a generic interval \([a, b]\), the scaling is computed as:

\[
T(p) = \frac{(b - a)(p - \text{min})}{\text{max} - \text{min}} + a
\]

(5.8)

With \(\text{min} = 0.25\), \(\text{max} = 0.75\), \(a = -1\) and \(b = +1\).

Equation (5.8) will be used in Section 5.5 to evaluate the experimental data collected on experiments and simulations.

While experiencing new interactions, the BN can acquire new statistical data and adapt its behavior over time. This happens when \(T\) changes sign, that is when the number of negative interactions surpasses the positive ones, or vice versa.
5.4 Experiment

5.4.1 Experimental Setup

As previously mentioned, for this experiment we used a Softbank Pepper, a humanoid social robot designed to operate in human environments. A single table is present in the environment, on top of which a printed mat, depicting the two positions $a$ and $b$ and some instructions for the participants, is laid down (see Figure 5.6). The robot is located on one side of the table, while the informants take turns in sitting in front of it, on the opposite side of the desk. The participants are provided with a sticker that they are able to move between the left and right locations. Each informant is instructed to act either as a helper, always revealing the correct position of the sticker, or a tricker, always giving wrong advice.

5.4.2 Procedure

As with the original experiment by Vanderbilt et al. (2011), our trial was divided in three sections: familiarization, decision making and belief estimation. Having already introduced the technical details of the cognitive architecture, here we will focus on describing the logical flow of operations. A video illustrating the following process can be found online (Vinanzi, 2019b).

5.4.2.1 Familiarization Phase

The aim of the first phase is to make the robot familiar with the informants, that means learning the correct parameters of the BNs associated to them. A visual description of the process is showed in Figure 5.7.

One at a time, each of the informants sit at the table while the robot captures some face images to be able to recognize them in the future. The user is given time to place the sticker on one of the two positions marked on the table, $a$ or $b$. After that, the robot asks for a suggestion on the location of the above-mentioned sticker and, once received, it follows it blindly, searching for the marker. Based on the results of this detection and on the maturity of the robot’s ToM model, an episode is created in its memory.

Following the original experiment, this demonstrative task is repeated 6 times per user, with the sticker located 50% of the time on position $a$ and the other 50% of the
CHAPTER 5. TRUST AND THEORY OF MIND

5.4.2.2 Decision Making Phase

In the decision making phase, shown in Figure 5.8, the robot has to correctly locate the sticker, choosing one of the two locations given the informant’s opinion. Initially, one informant sits at the table: if he or she is identified, the associated BN is fetched and used in the subsequent computations, otherwise a new one is generated on the fly for him or her using episodic memory.

Figure 5.7: Familiarization phase with a tricker informant. (a) The robot asks for a suggestion on the sticker’s location. (b) The informant places the sticker in one of the two positions. (c) The informant gives its suggestion on where to find the sticker. Note how the tricker gives misleading directions. (d) The robot searches for the presence of the sticker in the suggested position and records the episode.

time on position b. At the end of this procedure, the data acquired is used to build a BN for that informant.

The familiarization phase leads the robot into possessing a BN for every known informant.

5.4.2.2 Decision Making Phase
5.4. EXPERIMENT

Figure 5.8: Decision making phase with a tricker informant. (a) The robot asks for a suggestion on the location of the sticker and receives a misleading suggestion from the informant. (b) The robot performs inference on that informant’s belief network. (c) The agent decides that the informant will probably try to trick it, so it looks in the opposite location. (d) The robot finds the sticker and gives feedback to the informant.

At this point, the user positions the sticker and gives a suggestion. The agent performs inference on the BN in order to calculate the posterior probabilities given $Y_I$ as evidence. In particular, if $P_{Y_R}(a) > P_{Y_R}(b)$ the robot will investigate position $a$ and if $P_{Y_R}(a) < P_{Y_R}(b)$ it will look at position $b$.

The episode generated by this interaction will be used to update the parameters of the BN, making the robot progressively adapt to the user’ behavior.

5.4.2.3 Belief Estimation Phase

In the original trial, children were asked some meta-cognitive questions in order to investigate their perception of the informants and to examine their ToM matureness.
CHAPTER 5. TRUST AND THEORY OF MIND

To test the same on the artificial cognitive agent, Bayesian inference can be used. The belief estimation phase is very similar to the decision making one and differs only in the kind of inference computed. The robot uses its face detection and recognition algorithms to identify the informant with whom it is interacting, either a known or an unknown one, then it observes the table to identify the position of the sticker. Setting $X_R$ and $Y_R$ as evidence, the Message Passing algorithm is used to calculate the posterior probabilities for the rest of the network. At this point, the agent can use the probability distributions in nodes $X_I$ and $Y_I$ to infer the informant’s belief and the location that most likely would have been suggested by him or her. This process is shown in Figure 5.9.

5.4.3 Simulations on Character Formation

In order to test the effects of episodic memory on character formation, we created a simulated agent with a mature ToM and we enabled it to familiarize itself with different sets of informants to study how it would subsequently react to a novel person. Each set was composed of 8 informants. In particular, the first set was formed by 8 helpers, the second by 6 helpers and 2 trickers and so on until the last set was made up of 8 trickers. For each set of informants, 100 episodic belief networks were generated and their trust factors $T$ were computed using Equation (5.8) and plotted on a histogram.
5.5 RESULTS AND DISCUSSION

<table>
<thead>
<tr>
<th></th>
<th>Mature ToM</th>
<th>Immature ToM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DM</td>
<td>BE</td>
</tr>
<tr>
<td></td>
<td>a  b</td>
<td>a  b</td>
</tr>
<tr>
<td>$X_h$</td>
<td>0.5 0.5</td>
<td>1.0 0.0</td>
</tr>
<tr>
<td>$Y_R$</td>
<td>0.65 0.35</td>
<td>0.65 0.35</td>
</tr>
<tr>
<td>$X_T$</td>
<td>0.8 0.2</td>
<td>0.2 0.8</td>
</tr>
<tr>
<td>$Y_T$</td>
<td>1.0 0.0</td>
<td>0.62 0.38</td>
</tr>
</tbody>
</table>

Table 5.1: Bayesian network node values for both a Mature ToM (1) and an Immature ToM (2) agent. The rows of the tables represent the posterior probability distributions associated with each node of the networks, for both decision making (DM) and belief estimation (BE) tasks. In each setting, the informant is suggesting position $a$.

After each set was processed, the memory of the robot was reset so that the effects of each group of informants could be analyzed individually.

5.5 Results and Discussion

5.5.1 Trust and Theory of Mind

The final results of Vanderbilt et al. (2011) showed that only the children with mature ToM distinguished between helpers and trickers, thus confirming the fact that children’s reasoning about whom to trust is directly correlated with their understanding of mental life. Our research was coherent with these results: when the robot is provided with a mature ToM model, it is able to correctly recognize trustworthy sources from untrustworthy ones, accepting suggestions from the former while rejecting them from the latter. In contrast, if the agent is operated with an immature ToM model, it will fail in the evaluation. To understand why this happens, we will examine the posterior distributions in each node of a BN after the inference that takes place in the Decision Making and Belief Estimation phases.

Table 5.1 shows the results obtained from the interaction of a helper and a tricker with two agents with, respectively, a mature and an immature ToM.

In the decision making task, the helper indicated position $a$ when the sticker was placed in that location. The suggestion was accepted by both the agents, since $P_{Y_R}(a) > P_{Y_R}(b)$. The behavior of the agents differed towards the tricker which, in contrast, suggested position $a$ when the sticker was located in $b$. The mature agent rejected the suggestion and this happened because $P_{Y_R}(a) < P_{Y_R}(b)$, which means that the robot...
Figure 5.10: Reliability histogram of episodic belief networks generated by agents possessing different histories of interactions. Green bars represent trustful BNs ($T > 0$) and red bars depict BNs that tend to distrust ($T < 0$). Agents that have a more positive than negative background tend to be more prone to trust a new informant and vice versa. When $T = 0$ the informant is neither trusted nor distrusted and the agent will act randomly.

decided to look at the other location on the table. The immature agent could not recognize the deception and accepted the misleading advice, as observable in the posterior distribution $P_{Yr}(a) > P_{Yr}(b)$, thus falling for the deception.

During the belief estimation task, with the sticker placed in $a$ and a helpful informant, both the agents output $P_{Xi}(a) > P_{Xi}(b)$ and $P_{Yi}(a) > P_{Yi}(b)$, thus correctly anticipating the helper’s mental states. When facing the tricker, instead, the mature agent correctly predicts the malevolent intentions by outputting $P_{Xi}(a) < P_{Xi}(b)$ and $P_{Yi}(a) < P_{Yi}(b)$, while the immature agent failed doing so by predicting $P_{Xi}(a) > P_{Xi}(b)$ and $P_{Yi}(a) > P_{Yi}(b)$.

5.5.2 Episodic Memory

Given the non-deterministic nature of the algorithm used to generate episodic belief networks, a more statistical method of evaluation is needed to report the results of this module. The histograms of the trust factors $T$ obtained with the procedure described in Section 5.4.3 can be interpreted as the different characters that emerge in the robot, that
means the tendency it has to trust or distrust a novel informant based on the interactions it has been faced with in the past. So, as shown in Figure 5.10, an agent who is used to be tricked most of the time will tend to distrust somebody it meets for the first time, whereas a robot that has been treated kindly will learn to trust people and tend to consider them trustworthy until presented with contrary evidence. This behavior mimics exactly the “trust vs mistrust” stage of infancy theorized by Erikson (1993), in which children learn to shape their personality by succeeding or failing in developing trust based on the quality of cares received during infancy.

5.6 Future Directions

The model presented in this chapter estimates the trustworthiness of a collaborator through the personal experience of the robot, consisting of the current perception and the history of its interactions. In other words, it uses a unimodal perceptual information to produce a trust evaluation. We have plans to further extend this model to take into account other social cues from the human partner with the final objective to refine even more the trust estimation task. We argue that taking into account multimodal sensory information we will be capable of achieving a better trust evaluation which will be able to fit a wider range of scenarios.

Our hypothesis is backed by Cho et al. (2015), who promote the multidimensional nature of trust, with different kinds of factors affecting its evaluation. Of these, we are going to consider the ones which influence cooperation and collaboration in human-robot teaming, such as emotions (Ekman, 2009) and facial expressions (Boone and Buck, 2003), which can be integrated with other factors known as “macro-expressions”: symbolic gestures, tone of voice, demographic data and content of speech (Elkins et al., 2012).

Our plan is to extend our current architecture using an approach inspired by the Global Workspace Theory (GWT) (Baars, 2005). The latter is a cognitive model which is described metaphorically as a theater where several actors (the working memory) compete among them to earn the “spotlight of selective attention” on stage (the consciousness), while most of the background work remains invisible and behind the stages (the unconscious) (Baars, 1997). In more technical terms, the GWT model relies on the interaction between several expert systems that compete for the spotlight of the artificial consciousness. Figure 5.11 shows the proposed architecture based on GWT. The cognitive model hosts two expert systems: the Experience Expert System (ExES)
 CHAPTER 5. TRUST AND THEORY OF MIND

Figure 5.11: The proposed GWT model to predict trust in HRI. It is composed by two expert systems, both of which use different features to estimate the informant’s trustworthiness: the Experience Expert System (ExES) uses previous experiences of interactions, while the Engagement Expert System (EgES) makes use of social cues detected on the partner during the interaction. The final decision of the system will be moderated by the Global Workspace component.

and the Engagement Expert System (EgES), which are both capable of providing trust estimations. The Global Workspace (GW) component is in charge of deciding whether to assign the spotlight to one system or the other.

In particular:

- ExES is the trust estimation computational model as it has been described throughout this chapter.

- EnES is a cognitive model under development whose purpose is to estimate the trustworthiness of the informant using social cues identified from RGB images. These features, such as emotions extracted from the face and the voice tone, facial movements, micro-expressions and gaze direction are fused through an artificial neural network to produce an overall trust value in a self-supervised process which uses the result of the ExES to label the data it extracts (this means, associating a set of social cues to an estimated level of trust).

- GW, representing the artificial consciousness, will be a mathematical model whose purpose will be to select the results of either the ExES or the EgES to
apply to each specific situation.

This enhancement of the trust model is still under development and it is presented here as a highlight of one of the possible future directions in which we plan to aim our work on artificial trust estimation.

## 5.7 Conclusions

In this chapter we discussed an artificial cognitive architecture for trust, ToM and episodic memory in a HRI scenario that can enhance the performance of artificial agents in shared goal contexts. We have extended the previous work by Patacchiola and Cangelosi (2016) by integrating the original model into a complete robotic architecture and extending it with an episodic memory component which enables it to remember and make use of its past experiences to develop a personal character and, in doing so, to improve its cognitive abilities.

We strongly advocate the importance of computational models of trust in the current state-of-the-art of robot cognition, which has been currently mainly focusing on models of human trust towards robots. By virtue of this, we have plans to continue this line of research by further extending our architecture and applying it in more complex scenarios which will hopefully bridge the gap between a laboratory and a real-life scenario. One example of such was given in Section 5.6, but we also plan on taking into account the contemporary influence of two or more informants, similar to what has been done by Butterfield et al. (2009), to model robotic trust in heterogeneous multi-agent settings.
Chapter 6

Intention Reading and Trust for Collaborative Intelligence

6.1 Introduction

The past chapters of this thesis have focused on describing the motivations, implementations and empirical results of computational cognitive models of intention reading and trust for robotic agents. As stated at the beginning of this dissertation, our main objective was not to develop these cognitive skills as standalone implementations of human mental models for socially-aware robots, rather we consider them as the key factors to achieve artificial collaborative intelligence. Inspired by the statements from Bauer et al. (2008) and Groom and Nass (2007) who individually state the importance of intention reading and trust in a collaborative setting, the current chapter will present an integration of our two architectures to create a collaborative artificial agent able to direct its efforts in providing assistance to a shared activity with a human partner. This architecture is used in a scenario in which the robot will have to infer the goal of its partner by the observation of their social cues and subsequently perform decision-making to formulate an action plan that will try to optimize the chances of successfully achieving the intended objective.

Through the use of this computational model we aim at demonstrating the positive influence of trust on the synergistic efforts of the two agents. In summary, our main contribution comes in the form of a novel cognitive artificial architecture for HRC capable of performing both intention reading and trust estimations on human partners.
6.2 Integration of Intention Reading and Trust

As described in Figure 6.1, the main idea is that the trust model will act as a cognitive support for the intention reading, allowing the robot to fine-tune its behavior after having decided a general course of action. In particular, the robot will initially be trained on a set of goals and will thereafter try to understand which one is being pursued by its partner. Once a confident prediction is formulated, it will offer assistance in order to achieve the shared objective. The degree of help provided is influenced by the amount of trust the robot has in that specific person: if it thinks he or she have the appropriate knowledge or skills to complete the task, then it will act as an assistive peer, on the contrary it will start behaving more like a supervisor, observing more closely the partner, correcting their mistakes and, in general, assuming more of the responsibilities to ensure that the goal is eventually reached.

Before we describe how our intention reading and trust models can be integrated in a unified collaborative intelligence architecture, it is important to redefine the notation of the trust evaluation BN. The latter was introduced in Section 5.3.2 to determine the position of a sticker in one of two positions, but our current objective is to employ this model to check whether the human has the knowledge or skill to achieve a given goal: if this is not the case, the robot will have to perform corrective actions to ensure the success of the task. To do so, we have used the same network structure but we have changed the meaning of its binary nodes: \( a \) will represent a correct goal whilst \( b \) will symbolize an incorrectly executed goal. Following this convention, \( X_I \) and \( X_R \) will represent, respectively, knowledge of the informant and of the robot about the correct goals, \( Y_I \) will symbolize the choice of a correct or incorrect sequence of actions by the informant and finally \( Y_R \) will depict whether the robot should trust or not its partner.
The workflow we are about to discuss is shown graphically in Figure 6.2. The interaction starts with the robot trying to identify the partner: in case of success it fetches their trust belief BN, otherwise it generates one on-the-fly through its episodic memory. The robot is assumed to naively trust the person that has trained it, so it will possess at least one BN in its memory. After this process is completed, it will start observing the human to read his or her intention. Once a goal is predicted, the robot will perform a proactive trust evaluation in which it will ask itself if it expects the human to fail or succeed in the task at hand: this process is equivalent to the Belief Estimation task which was described in Section 5.4.2.3. Based on this evaluation, the robot will adopt one of two different approaches.

If the robot decides to trust the human, then it will collaborate towards the achievement of the predicted goal. Once the task is complete, it will judge the Total Output (TO) of the joint action: if the shared effort led to a successful, valid outcome its trust level towards the partner will increase, on the contrary it will decrease. If however the robot decides to distrust the partner, it will immediately inspect the Partial Output (PO), that is the portion of the task that has already been completed before an intention prediction was formulated. If the PO appears invalid, the robot will lower its trust level and will thereafter try to correct the mistake and take over the rest of the task. If instead the PO is a valid one, even if not the one which the robot had predicted, the agent will give the partner a chance to regain trust by collaborating and evaluating the TO, as described previously.

This workflow penalizes human partners who are both incapable or unwilling to contribute with an appropriate effort to the shared task, but at the same time gives distrusted people a chance to regain the trust of the robot. This is important, because failures could arise from temporary situations such as injuries or fatigue, in which case opportunities to compensate should be available.

In an effort to include some features of Explainable AI (Hagras, 2018) into our system, the robot will try to be transparent and constantly communicate to its human partner any estimation results and any changes in its levels of trust. So, for example, if the robot doesn’t trust the human to be able to accomplish a pursued goal, it will state that clearly, thus justifying its much more strict behavior. In particular, the robot will always state: the predicted goal, the estimated trust levels including any changes from trust to distrust or vice versa, its evaluation of the TO or PO and the explanation of why it believes that a task was unsuccessful. Finally, the agent will also try and justify its own errors: for example, if it realizes that the achieved goal was not the predicted
Figure 6.2: The collaborative workflow. The robot reads the intention and decides whether to trust its partner or not. In the former case, it provides assistance and evaluates the total output resulting from the collaboration, otherwise it adopts a more strict supervision on the human: if the partial output seems valid it gives them a chance to regain trust, otherwise it will take over the task and attempt to correct the mistakes.

one, but nevertheless was valid.

6.3 Experiments

6.3.1 Experimental Setup and Procedure

Having already validated the performance of both the intention reading and trust models, the aim of this experiment is to verify our hypothesis on the positive influence of trust mechanisms on the overall collaborative performance. We will therefore use the
same block building experimental setup presented in Figure 4.5 and compare the results achieved from our new, integrated architecture (referred to as Trust-Architecture, or TA) with the baseline obtained from our previous intention reading model (the one presented in Chapter 4 which we will hereafter be referring to as the No-Trust Architecture, or NTA).

As in the previous experimental procedure, the human will demonstrate each of the 8 goals once and the robot will learn to associate them to the perceived social cues. At the end of the training, the agent also initializes a trustful BN for the demonstrator: this is done because we want it to trust the person who provided its instruction. During the execution phase, the robot will follow the workflow described in Section 6.2. In our setting a TO represents a full line of 4 colored blocks, while the PO is the sequence of cubes that the human has arranged before the artificial agent was able to perform intention reading. If the human is trusted or the PO is valid, the robot will collect the next predicted blocks and hand them over to him or her. If not, the robot will position the blocks itself on the building area in what it considers to be the correct order, attempting to correct the errors that have been committed. In the latter case, the robot will also offer an explanation of why it thinks the PO is invalid (in our experimental setting, this happens when two blocks from the same side of the table are placed one next to another).

In the scope of this experiment, an interaction will be considered successful if its outcome is a structure that follows the game’s rules, in other words one of those listed in Figure 4.5b. This is true even if the true goal was not the one predicted by the robot: this is because we do not wish to measure the performance of the intention reading model (which we have already quantified in Table 4.2) but rather we want to evaluate the collaborative effort itself. From here on, we will define a “positive” interaction one in which the human correctly achieves a valid goal and a “negative” one where he or she takes an unsuccessful course of action. The human might violate the rules more or less intentionally, but for our purposes we consider both these cases as a failure that will lead to a decrease of their trust level.

### 6.3.2 Simulations

To verify and measure the trust model’s impact on the collaborative effort driven by the intention reading architecture, we have conducted a batch of simulated experiments\footnote{The use of virtual agents in a simulated environment is a COVID-19 lockdown contingency choice.}
using a virtual robot which has been modeled in accordance to the empirical data collected during our latest experiment on intention reading. In particular, we have provided this agent with an observation error obtained from the confusion matrix reported in Figure 4.9 generated by 160 real interactions. We have not included an actuator error rate because in our previous experiment the robot was able to operate flawlessly on the tabletop scenario.

After training the robot, we let it interact with a set of simulated humans which possess different behavior patterns. It is important to note that in most of these experiments we do not make an explicit use of episodic memory. This is because, having only familiarized with the demonstrator, the robot would generate a fully trustful network for the novel informant because it will be sampling episodes from a batch of positive memories. This mean that, for the purpose of the simulated experiment, we can simply assume that the robot will naively trust its new informant. Thereafter we continue not using the memory system because we do not want our results to depend on the order in which the robot has experienced the users, rather we want to study how each robot would respond to each user independently. For completeness, one of our simulated humans is initialized with a distrustful BN to simulate the effects of the episodic memory.

We have divided the simulated humans in two groups, which are graphically represented in Figure 6.3. The first one involves the “deterministic” agents, which have a fixed behavioral pattern, such as:

- $H_1$: always negative;
- $H_2$: 50% positive, then 50% negative;
- $H_3$: 50% negative, then 50% positive;

The second group categorizes the “stochastic” agents: the latter possess different success-to-failure ratios, but the order of their actions is randomized and not fixed. In particular, we have:

- $H_4$: 50% success-to-failure rate;
- $H_5$: 80% success-to-failure rate;
- $H_6$: 20% success-to-failure rate;
- $H_7$: 80% success-to-failure rate, but initialized with a distrustful BN;
Figure 6.3: A graphical representation of the simulated humans’ behavioral patterns. The deterministic agents are defined by the fixed sequence of interactions over time that they perform, whilst the stochastic group members are characterized by the success rate of their actions, which is then randomized at every iteration.

The deterministic humans have been tested through a batch of 100 iterations each. For the stochastic ones, we have performed 10 random initializations and for each of them we have executed 100 interactions with the simulated robot. The only exception is $H_4$, for which we performed 20 random initializations due to its high variance. During each test we have recorded the success rate and the opinion value, both of which are described in the following section.

### 6.3.3 Evaluation Metrics

#### 6.3.3.1 Success Rate

Given a human partner $H_i$, we define the success rate $S$ of an interaction as:

$$ S(H_i) = \frac{\text{successful goals}}{\text{total interactions}} \in [0, 1] $$

We wish to formulate a comparison between the integrated cognitive architecture
6.3. EXPERIMENTS

and the NTA. To do so, we refer to the success rate calculated on the latter as $S^*(H_i)$ and we formalize the difference between the two systems as:

$$\Delta S(H_i) = S(H_i) - S^*(H_i) \quad (6.2)$$

Positive values of $\Delta S(H_i)$ will denote a more performative collaboration obtained by our current architecture over the NTA and vice versa.

6.3.3.2 Artificial Opinion

We define a quantitative index which reflects the willingness of the robot to change its opinion about a partner. This is directly correlated to the trust factor described in Section 5.3.4 but offers a better graphical understanding on how many observations the robot is willing to observe before switching between states of trust and distrust. For a partner $H_i$ at a certain time step $t$, this artificial opinion is calculated as follows:

$$O(H_i, t) = \frac{n_p - n_n}{n_p + n_n} \in [-1, 1] \quad (6.3)$$

Where $n_p$ and $n_n$ indicate respectively the number of positive and negative episodes experienced by the robot with partner $H_i$ at time $t$. We will sometimes use a more simple notation, where we indicate the opinion of a robot towards a generic partner at a certain timestep simply as $O(H)$.

When the robot trusts the person, that is when $P_{X_i}(a) > P_{X_i}(b)$, it is also true that $O(H) \geq 0$ and vice versa, when the BN is distrustful towards them $O(H) < 0$. The choice of having the robot to trust when $P_{X_i}(a) = P_{X_i}(b)$ and $O(H) = 0$ is made by design, since we wish the robot to act more friendly towards its users, giving them the benefit of doubt. The closest $O(H)$ is to 0, the easier it will be for the agent to flip its trust and vice versa, the more this value tends towards the extremes, the less inclined the robot will be to alter its belief. Of course, $O(H) = \pm 1$ indicates a very strong opinion and it is possible only when the agent has experienced solely positive or negative episodes with that specific user.
6.4 Results and Discussion

6.4.1 Success Rates

In our last experiment on intention reading, presented in Chapter 4, we have considered partners which always act towards one of the correct goals. This means that for a hypothetical human $H_0$ acting always positively, $S(H_0) = S^*(H_0) = 1.0$ despite the fact that the empirical results reported in Table 4.2 indicate that the robot succeeds 80% of the time: this is because in the current investigation we are not testing the intention reading capabilities, which enable the collaboration in the first place, rather we want to analyze the effect of a trust mechanism to correct partners who are not capable or willing to achieve a valid goal.

However, if we start considering humans which can (more or less intentionally) fail the task, the NTA’s success rate drops drastically as it does not possess the ability to adopt any corrective actions. In this case, each action failed by the human will result in a failed collaboration. Figure 6.4 shows a comparison between the success rates of the two architectures measured on the 7 simulated humans. $H_1$ always fails the task so $S^*(H_1) = 0$, while the trust-enabled model is able to score $S(H_1) = 0.97$ with a
Table 6.1: Mean and standard deviation of the success rates calculated on the interactions performed by the stochastic simulated humans.

<table>
<thead>
<tr>
<th>Partner</th>
<th>Mean (µ)</th>
<th>Standard deviation (σ)</th>
<th>Initializations</th>
</tr>
</thead>
<tbody>
<tr>
<td>H₄</td>
<td>0.66</td>
<td>0.15</td>
<td>20</td>
</tr>
<tr>
<td>H₅</td>
<td>0.8</td>
<td>0.0</td>
<td>10</td>
</tr>
<tr>
<td>H₆</td>
<td>0.95</td>
<td>0.01</td>
<td>10</td>
</tr>
<tr>
<td>H₇</td>
<td>0.81</td>
<td>0.01</td>
<td>10</td>
</tr>
</tbody>
</table>

significant increase of $\Delta S(H_1) = 0.97$.

Both $H_2$ and $H_3$ provide a mixed scenario in which the behavior of the simulated human is quite regular by being respectively positive and negative for half of the time, in inverse order. In both these cases, the NTA could only score $S^*(H_2) = S^*(H_3) = 0.5$.

The trust mechanism did not prove itself of much use for $H_2$, since the robot builds up a strong trust for the user and is not able to change its mind in time to correct the new behavior: as we will see in the next section, this is because the agent should be observing at least $n_p + n_n + 1$ negative cases to completely change its mind about the informant, which is not possible in this 50-50 split case initialized with positive episodes. In summary, $S(H_2) = 0.5$ and $\Delta S(H_2) = 0$, in other words the performance is the same as the one obtained through NTA. $H_3$ behaves similarly: not having enough time to change its mind, the robot continues to distrust the human nearly until the end. The difference is that in this condition the robot maintains a strict supervision on the interactions, leading to $S(H_2) = 0.97$ with an increase of $\Delta S(H_3) = 0.47$.

To better evaluate the stochastic humans, we have recorded the success rates achieved through the batches of random initializations and we have calculated the mean $\mu$ and the standard deviation $\sigma$. The success rates reported in Figure 6.4 for these simulated people represent the mean score, supplied with error bars representing $\sigma$. These values are also recorded in Table 6.1 for better visualization.

$H_4$ is the agent who achieved the highest $\sigma$, that’s because its behavior is the most unpredictable. This is explained by considering what this behavioral pattern represents: with 50 positive and 50 negative episodes with randomized order of appearances, the trust levels can fluctuate significantly. This is also the reason behind our decision to execute double the number of trials with this simulated human. In this case, the NTA would have achieved $S^*(H_5) = 0.5$, but the trust-enabled architecture is able to score $S(H_2) = 0.66$, with $\sigma = 0.15$, achieving on average $\Delta S(H_4) = 0.16$. The performance of the TA has a theoretical lower bound equal to the one obtained by the NTA and
in fact we have registered scores per batch not lower than 0.5, up to a maximum of 0.93. We can conclude that a success rate of 50% is a critical point of uncertainty in which the human’s behavior is too variable for the robot to adapt efficiently. As we will see shortly, above this value the human becomes more skilled and the value of trust-based corrective mechanisms gradually fades away and vice versa, lower success rates benefit more from the TA.

$H_5$ is a fairly expert human who succeeds 80% of the time, which means that $S^*(H_5) = 0.5$. The robot builds a very solid trust towards this partner, at the point that the 20 failures are, in our experiments, sufficiently sparse in the set of 100 interactions to never make the trust flip to negative. The latter is of course theoretically possible, but they should appear clustered at the beginning of the batch to make that occur. This means that the robot never loses trust towards this confident human, but that also those 20% failures are not being captured, hence $S(H_5) = 0.8$, $\Delta S(H_5) = 0$ and $\sigma = 0$. This result is quite important because, as we mentioned previously, it confirms the fact that the effectiveness of trust evaluations on the collaboration are inversely proportional to the skill of the partner.

The behavior of $H_6$ is quite the opposite of $H_5$, succeeding only 20% of the times. In this case, $S^*(H_6) = 0.2$ but the full architecture was quickly able to detect the negative attitude of this simulated human and it promptly started distrusting them, achieving $S(H_6) = 0.95$ with $\sigma = 0.01$, leading to an average $\Delta S(H_6) = 75$.

$H_7$ has the same behavioral pattern than $H_5$, which is an 80% success rate, but the robot facing him or her is not initialized with a trusting BN but rather with a naively distrustful network. This is meant to test the effects of the episodic memory on the performance of the architecture. As we can see from Figure 6.4, we achieve a similar result as $H_5$, just slightly better because the robot will tend to not trust them and take over the task until it is persuaded about their skill. The mean result for this scenario is $S(H_7) = 0.81$, with $\sigma = 0.01$ and $\Delta S(H_6) = 0.1$. What this results stands for is the fact that the episodic memory has only a local effect on the robot’s behavior, which is tuned on the long-term through real interactions which take over its initial prejudice.

Overall, the experiments showed an average success rate increase equal to:

$$\frac{1}{7} \sum_{i=1}^{7} \Delta S(H_i) = 0.33$$  \hspace{2cm} (6.4)

Thus confirming the positive impact of trust estimation in support of intention reading during collaborative HRI.
6.4. RESULTS AND DISCUSSION

Figure 6.5: Variation of the opinion value at each turn of interaction for the 3 deterministic simulated informants ($H_1$, $H_2$ and $H_3$), initialized with a trusting BN. When $O(H)$ becomes less than 0, the robot starts distrusting the informant and taking more control of the task.

### 6.4.2 Trust Dynamics

During the simulated interactions, we have recorded the opinion value for each of the human partners. By a design choice, the robot is initialized with a trustful BN built from 4 positive episodes. This network yields an initial opinion $O(H, 0) = 0.4$. After that, we recorded $O(H, t)$ for $t \in \{1, 100\}$ and we reported them in a set of graphs.

Figure 6.5 shows the dynamics of the robot’s opinion through the various interactions for the deterministic humans. $H_1$ always acts incorrectly, but the network is initially willing to trust them. This changes very quickly, since we can observe the opinion dropping to 0 after only a few negative episodes and then decreasing close to the lower limits. This value never actually reaches the minimum value of -1 because this would only be possible if the robot had experienced 100% negative episodes, which is not the case due to how its BN was initialized. In any case, we can see how the opinion of the robot stays low, meaning that the human will have to put a lot of effort to regain its trust.

$H_2$ behaves 50 times positively and, subsequently, 50 times negatively. During the first half of the interactions the opinion raises to its maximum, since the robot has only
experienced successful interactions with that person. From turn 51 onward, the human starts failing the block building game and the opinion slowly decreases but it is not able to flip. This is because, by the end of the session, the robot possesses 54 positive and 50 negative episodes in its memory, meaning that it has not enough time to change its mind (another 4 negative episodes will bring the opinion to 0 and another one after that will flip the trust).

$H_3$ acts in the opposite way as for the previous simulated agent. The trust quickly drops towards the distrusting side of the graph and slowly rises after turn 50. In contrast with $H_2$, this human is able to flip the trust back to positive by the end of the session, because of the way it was initialized. If the BN was originally set to distrust, these two graphs would become inverted.

As previously mentioned, the random nature of the stochastic humans required several batches of iterations, performed with different random initializations, to fully understand the behavior of each agent.

Figure 6.6 reports the dynamics of the robot’s opinion during 10 out of the 20 iterations performed for $H_4$, which is the simulated agent with a success rate of 50%. What is immediately noticeable from these graphs is that the opinion always converges.
6.4. RESULTS AND DISCUSSION

Figure 6.7: Opinion dynamics for the stochastic human $H_5$ (80% success rate) during 10 iterations.

around 0: this is an expected result, since this value is the midpoint in the scale, representing partners with mixed, indecisive behaviors. It is worth remembering that the robot will trust a human when $O(H) \geq 0$.

Regarding $H_5$, having a success rate of 80%, we expected the robot to terminate each iteration with a high opinion. This prevision was confirmed by the graphs reported in Figure 6.7, which show that the robot never fully changed its impression of the partner, in other words the 20 errors randomly scattered among the 100 interactions were not sufficient to flip the trust. The closest that the robot felt distrust in them happened in the first diagram, where a sequence of negative episodes were experienced right at the beginning, dropping the opinion to 0, which by our design still represents a trusting situation.

Similar considerations are valid for $H_6$, the virtual agent capable of a very low, 20% success rate. The 10 diagrams of Figure 6.8 differ mostly on the very first interactions, when a sequence of positive episodes may impact the limited memory of experiences of the robot and in fact some of the iterations have managed to achieve trust for some turns. Ultimately, the opinion always ends up settling on the lower side of the graph, in the distrust domain, which is what we expect from a human who consistently fails the majority of the tasks.
CHAPTER 6. COLLABORATIVE INTELLIGENCE

Figure 6.8: Opinion dynamics for the stochastic human $H_6$ (20% success rate) during 10 iterations.

All the previous simulations have been executed on a simulated robot initialized with a trustful BN, for the reasons we have explained in the preceding sections. We now wish to analyze what would happen if the network was created through episodic memory, that is if it does not contain 4 positive episodes but a certain number of negative ones. For this reason, we have built $H_7$ with a BN composed of 4 negative episodes: this yields $O(H_7, 0) = -0.4$. Figure 6.9 shows the result of this experiment, which is comparable to the one performed for $H_5$ since these two simulated humans behave in the same way, with the only difference being the initial prejudice. Despite the variance in the early interactions, which can make the opinion oscillate quite widely, on the long run the latter settles for similar values registered for $H_5$. This demonstrates that the episodic memory can create a local effect which influences strongly the early interactions of the robot with a person but that fades gradually once the actual experience takes over the initial prejudice. This is exactly how the episodic memory system was intended to operate. Having tested the two types of BN that can be generated by the episodic memory system (completely polarized towards trust or distrust) we do not feel the need to investigate the cases which lie in between: these will produce similar, but more mitigated, effects than the ones we have observed.
6.5. CONCLUSIONS

In this chapter we have described the experiment that brings together the foundational work which has been discussed throughout this thesis. Our final cognitive architecture for this research project lays its foundations on our models of intention reading and trust and integrates them to obtain a computational model of developmental collaborative intelligence for embodied robotic agents. Through this architecture, a social robot is able to guess the unspoken intention of a human partner and decide the best course of action to help him or her achieving their objectives. We have mathematically verified how the inclusion of trust considerations in teaming situations can significantly improve the outcome of the joint action, especially when the human partner does not possess the skill or ability to execute a correct effort, leading to an overall higher success rate.

Despite this result, the author feels there is space for improvement. For example, in this work we are not considering the difference between a human which lacks the skill to accomplish the task versus one which intentionally causes a failure. Whereas the former would be treated by the robot in the way in which we discussed in this chapter, the latter would require additional consideration, not only from a technical

Figure 6.9: Opinion dynamics for the stochastic human $H_7$ (80% success rate, against a naively distrusting BN) during 10 iterations.

6.5 Conclusions
point of view (a deceit has to be distinguished from a genuine error) but also from a philosophical perspective. Specifically, if a person is intentionally trying to hinder the robot, why should the machine be willing to collaborate? At a glance it would appear as if the robot would do so only if it had a personal commitment towards the task, for example if another human has ordered it to complete a certain job, but further investigations are required to properly address the problem. This may well be the subject of future investigations.
Chapter 7

Conclusions

7.1 Overview

Collaboration between people has been, through history, the key to obtain the grand achievements of the human species. In a future world where humans and robots will be living closely, we want collaborations between the two to be fluid and natural. With this purpose in mind, we state that a true collaborative robot able to operate in human-sized environments must possess the same cognitive skills that drive our own social life. In this thesis, we defined collaborative intelligence as the mutual interaction between intention reading and trust estimation: two mental abilities towards which humans are biologically inclined. The former allows an agent to understand the actions and goals of other agents acting around it, thus providing clues and meaning to simple sensory perceptions, while the latter is essential to estimate the level of skill or knowledge of another agent so as to formulate appropriate decisions. Following the developmental robotics principles, our cognitive architecture takes inspiration from scientific findings in human cognition and both the intention reading and trust models are designed accordingly to the current psychology literature. We have developed a cognitive system which is able to learn goals by demonstration in an unsupervised and probabilistic way and to estimate trust using an artificial ToM. We have applied this architecture to a block building game where a robot is engaged with one or more humans to pick and place some colored cubes from a table to form constructions that follow certain patterns.

Overall we can conclude that the synergistic combination of intention reading and trust leads to better results than the ones obtainable by just predicting the human’s goal. The experiments that we conducted have shown that the complementary use of
both these cognitive skills has the potential to enhance the collaborative performance, making the robot act as a better teammate. This confirms our initial hypothesis, which is that collaborative intelligence is enabled by the ability to read another agent’s intention and is fostered by the capacity to correctly estimate the trustworthiness of the other party. The robot’s ability to take control of the task whenever the partner demonstrates a lack of skill results in a significant increase in the success of the joint task.

7.2 Summary of the Contributions to Knowledge

This section will revisit the research questions and expected contributions formulated in Chapter 1 in the light of what has been discussed throughout the previous chapters. The key goal of the work presented in this thesis was to advance the scientific understanding of trust, intention reading and collaborative intelligence in the interaction between human and robots through the design of a robot learning architecture based on the developmental robotics approach, other than to test our hypothesis on collaborative intelligence.

Here we reiterate the research questions introduced in Chapter 1 and provide an answer to each of them based on the work discussed throughout this thesis:

RQ1. *Can a robot learn to infer intentions from a human partner through the observation of behavioral and social cues, using lightweight and unsupervised methodologies which do not involve large datasets, long training times or hand-crafted goal libraries?*

A robot is able to learn to infer intentions from a human partner through the observation of a range of social cues. In particular, we have proved that body posture and gaze direction, which are essential sources of information for developing children, can be just as valuable for an artificial agent. Our methodology involves a low-level unsupervised action encoding paired with a high-level probabilistic goal prediction which doesn’t involve the use of large datasets (the robot only needs to observe each goal once), is fast to train and uses no hand-crafted plan libraries. We have shown how a robot can autonomously learn how to cluster the partner’s physical configurations and analyze how their actions transition between these clusters to anticipate the desired end objective fast enough to offer its assistance. By using only RGB cameras to collect visual information from the human, and not relying on RGBD sensors, external hardware or wearable
7.2. SUMMARY OF THE CONTRIBUTIONS TO KNOWLEDGE

devices, we have not bound our cognitive architecture to a range of robots possessing a certain hardware configuration but we have extended it to the majority of platforms available in the current state-of-the-art. To prove this, the experiments described have been performed on several, structurally different robotic platforms. Publications 2 and 4 refer to this contribution.

RQ2. Trust is a critical issue in HRI, but it has mainly been studied from the human’s perspective. What would happen in the reverse scenario, where a robot is the trustor and a human is the trustee? Can a robot learn to dynamically estimate the trustworthiness of a human partner and act based on that prediction?

Robot-centered trust considerations can be valuable during HRC interactions where each member of a team must rely on the actions of the others to accomplish a shared goal. With this in mind, we have shown how it is possible for a robot to estimate the trustworthiness of another agent, using this prediction to guide its own actions. The strong correlation between trust and ToM has been underlined and consolidated through a Bayesian model inspired by a psychology experiment, which has been encapsulated into a robotic architecture and expanded with an episodic memory system. The latter is necessary for the formation of the personal character of the robot and allows it to naively trust or distrust strangers based on its past experiences. Publications 1 and 3 are related to this contribution.

RQ3 Based on current psychological theories, we state that collaborative intelligence arises from the mutual interaction of two cognitive skills, namely intention reading and trust estimation. Is this true? Can a robot use these faculties to assist a human partner? Does a collaboration involving trust perform better than one based solely on goal prediction?

We have demonstrated that it is possible for a collaborative intelligence to emerge from the mutual interaction of intention reading and trust estimation cognitive capabilities in an artificial agent: the former is used to determine which type of assistance is needed by the partner, whilst the latter is used to fine-tune the robot’s behavior in order to optimize the chances of success. Through a set of simulated experiments modeled by empirical data we have demonstrated that a collaboration involving trust estimations performs better than one which relies only on intention prediction. This result is significant because despite the renown value of trust in human interactions, present collaborative models in HRI fail, to
the best of our knowledge, to consider this factor from the robot’s point of view. Publication 5 is related to this contribution.

Our overall achieved contributions are the following:

- An artificial cognitive architecture for intention reading in HRI which is inspired by developmental principles (Tomasello et al., 2005) and is able to perform lightweight, fast and unsupervised mind-reading on a human partner.

- A novel multimodal clustering algorithm, FSSC, which performs multiple clustering operations in several feature-spaces to progressively refine its partitioning process, spacing out data which would lie closely on the data space.

- A cognitive architecture for trust estimation in HRI where the robot is the trustee and the human is the trustor, validated against a developmental experiment (Vanderbilt et al., 2011) and enriched with an episodic memory system to simulate the development of an artificial personality.

- A collaborative intelligence architecture which emerges from the synergistic coupling of intention reading and trust, which is able to outperform our previous HRC model thanks to evaluations on the robot partner’s trustworthiness.

7.3 Limitations

This section provides an overview on some of the limitations of the work presented in this thesis and discusses some possible solutions. The previous chapters have already highlighted some of the limitations, which will be reported here for the reader’s convenience.

- With reference to the intention reading model, the robot is not considering in its evaluations some environmental factors such as which objects the partner is manipulating, rather it only keeps into account the movements which are being performed. As discussed in Section 3.6, the inclusion of this type of information is likely to provide additional clues for the robot’s social understanding of the observed scene. The author’s opinion is that this problem could be addressed by combining visual object detection and recognition with QRS descriptors to model the dynamics of the environment.
• In a real-life scenario, one goal could be achieved through different sequences of actions. Our current intention reading model does not keep this into account and associates each intention with one sequence of actions which lead to its achievement. This limitation might be overcome by teaching the robot a different set of actions that lead to the same goal label (or a variation of such), but this would not entirely solve the problem. Tennie et al. (2006) state that a behavior is composed of actions (the motor patterns used by the demonstrator to achieve their goal) and results (the changes produced in the environment by those actions). Imitation is the act of learning how to mimic the actions that have been observed, whilst emulation is the ability to reproduce the results. In other words, an agent that learns to emulate is able to generalize the goal to a wider set of possible actions. This problem falls under the domain of robot emulation learning (Ragaglia et al., 2018) and would require further investigation in order to be solved.

• In our trust model, the evaluation is purely performed on the basis of experience. This means that the robot will not consider any other factor, such as social cues from the human it is interacting with. Cho et al. (2015) state that trust is a multidimensional variable that depends on a range of logical and emotional factors. From this, we can conclude that measuring a level of trust based solely on a unimodal evaluation might be too restrictive for real-life scenarios and multimodal computations must be adopted. This might be addressed using the methodology discussed in Section 5.6, in which we propose an artificial model of consciousness that mediates between experience and social cues.

• Due to the inability to access appropriate research facilities due to COVID-19 lockdown in the United Kingdom, it was not possible to perform experiments on the physical robot regarding the full collaborative architecture discussed in Chapter 6, which were instead replaced by simulations. By providing the virtual robot with the same empirical error rates obtained during the foundational experiments, which have all been executed in the real world, we have tried to minimize any approximation errors between the simulated and the real interactions.

7.4 Future Work

The work described in this thesis opens up several opportunities for future investigations. The following lines of research are the most appealing and promising directions
which we believe can have an impact in the field of cognitive robotics for HRI.

One of these directions focuses on overcoming the limitations of our current models, as discussed in the previous section. For reference, these consist in making the robot aware of its surroundings and about the spatial relationships between elements of the environment to perform a better intention reading, embedding elements of emulation learning to make the robot able to generalize the actions that lead to goals and expanding the trust model to take into account social cues other than mere experience when formulating a prediction on someone. All of these enhancements promise to widen the social awareness of the collaborative robot and to close the gap between a laboratory setting and the real world. Some work is already in progress on the trust model, as described in Section 5.6.

Another interesting research opportunity arises from the possibility to enrich the intention reading model with hierarchical goals: instead of having a single low-level and high-level structure, it could be possible to include several orders of high-levels to infer more complex goals and meta-goals. This would allow the robot to identify several sub-goals which could lead to a higher goal that is in turn driving the actions of the partner. Anticipating the latter would allow the robot to collaborate even better, as it will provide a guideline on the next future actions with a possibly large advance margin.

The most promising future direction is the one of multi-agency: many HRI studies have covered the synergistic behavior and mutual understanding of two agents involved in joint action and the literature presents several computational architectures able to model a dual interaction, but scenarios involving a multitude of cooperating agents still present many challenges and unexplored opportunities (Dorri et al., 2018). Multi-agent systems (MAS) is a sub-field of artificial intelligence that deals with the computational interaction between two or more intelligent agents involved in a shared task. We can imagine a MAS as an ensemble of agents, each of which has the autonomy to contribute to a problem-solving network. In our planned future work, we are particularly interested in heterogeneous teams constituted by a mixture of humans and robots, involving at least three different agents. In these kinds of scenarios, a robot will have to predict and adapt to the behavior of the other members of the team, be these human users or other robots. In particular, we are interested in modeling collaborative intelligence, taking inspiration from our cognitive models of trust and intention reading. The successful investigation of this topic will result in significant advances on
the scientific understanding of trust and intention reading in the interaction of teams of humans and machines for complex scenarios, and the development of machine learning and robotics algorithms for trustworthy cooperation in teams of heterogeneous and distributed agents.

In a novel scenario involving a multitude of heterogeneous agents, new research questions arise:

1. How can individual agents identify the overall Team Goal, which results from the collaborative and complementary action of individuals and sub-teams?

2. How will the performance of sub-teams and their sub-goals be integrated towards the achievement of the Team Goal?

3. Which kind of behavioral and social data is required to adapt a dual collaborative model to a triad of agents? How will this scale to a network of \( N \) agents?

4. Which are the cognitive mechanisms that have to be adopted to accommodate this architecture to a distributed, as opposed to local, network of agents?

To answer these questions, several sensor modalities will have to be considered for each agent, including but not limited to: behavioral (postural configurations, gaze directions), social (mutual interactions between agents) and language (both explanatory and descriptive utterances) interaction data.

The MAS scenario offers the opportunity to conduct research not only on intention reading, but also on trust estimation. In particular, it would be interesting to investigate how a robot could model the trustworthiness of a team. For example, in order to provide the best collaborative effort the robot might have to choose another agent to assist it in a certain task. The optimal choice would be, of course, the team member which is more trusted on that specific job. In other words, the robot should maintain a belief system about its team composed as such:

\[
T(i, j, \alpha)
\]  

(7.1)

This notation, introduced by Marsh (1994), denotes the level of trust given by an agent \( i \) to another agent \( j \) in a situation \( \alpha \). Sometimes, especially in distributed teams, an agent does not possess direct experience on all other agents and will have to rely on the recommendations of its acquaintances (Sapienza and Falcone, 2020).
The author is confident that further research on the above areas will enrich the state-of-the-art in cognitive robotics with novel methodologies and insights inspired by the work which has been described in this thesis.

7.5 Epilogue

The themes discussed throughout this thesis underline the importance of embedding modern and future cognitive robots with mental mechanisms that will allow them to act naturally in environments shaped by human-made social attitudes. Being able to create intelligent machines capable of collaborating smoothly by autonomously understanding our intentions and evaluating our chances of success is a fundamental step towards the adoption of widespread robotic agents in the real world.

By combining machine learning, robotics, probability theory and developmental psychology, our work showed promising advancement in this area and enriches the state-of-the-art of intention reading, trust, Theory of Mind and collaborative intelligence computational modeling. This will hopefully inspire new, promising research in this direction that could open up exciting new opportunities in the near future.
Bibliography


Appendix A

Decomposition of the Intention Reading Goals in the Constituting Skeletons

In Section 3.4 we have described the experimental setting for our pilot experiment on intention reading, involving a dynamical clustering of the human body postures. In this appendix we analyze the training set that was used to build the knowledge base of the robot. For each of the four goals, we report the skeletons of the visible upper body of the teacher that, during the demonstration, have triggered a cluster transition. We accompany these graphs with each skeleton sequential id and the cluster id to which they have been assigned by the system. Note how the skeleton ids do not progress linearly, since Transition Analysis filters out from the encoding stalls in the same cluster to achieve temporal invariance. For each of these goals, it is possible to see the human demonstrator beginning in a central stance, then reaching either to the left or right, bringing the collected block to the center and so on.
Figure A.1: Skeleton decomposition for the goal ‘wall’.
Figure A.2: Skeleton decomposition for the goal “tower”.
Figure A.3: Skeleton decomposition for the goal “castle”.
Figure A.4: Skeleton decomposition for the goal “stable”.
Appendix B

Intention Reading Training Sets

We illustrate here the training sets that have been provided to the robot during the two intention reading experiments described respectively in Chapter 3 and Chapter 4. The first one is recorded through the eyes of the iCub robot, which collected images with a resolution of 320x240 pixels, while the second consists of frames retrieved from the Sawyer’s head camera in a 800x600 resolution. These sequences of human actions are recorded during the training phases of the two experiments and are used by the cognitive architectures to extract social cues (skeleton keypoints for the first, joint coordinates and gaze direction for the second) that will be used in the low-level action representation and encoding.

Given the much larger number of goals included in the second experiment, we have decided to display only one of them, which in any case is sufficient to illustrate the experimental procedure that was adopted.
Figure B.1: Training set for the intention reading experiment described in Chapter 3.
Figure B.2: Training set for the “BGOR” goal from the intention reading experiment described in Chapter 4.